Design of the Reverse Auction in the Broadcast Incentive Auction

Peter Cramton, Hector Lopez, David Malec and Pacharasut Sujarittanonta

15 June 2015 (first draft 12 March 2015)

Appendix: Detailed description of the simulation model and analysis

Contents

System architecture

Performing large-scale simulations of the incentive auction and analyzing the results involves a wide range of software and hardware resources. In addition to running simulations on local machines, we harness the power of the cloud to achieve simulations at a much larger scale than would have been otherwise possible. Scaling up the simulations, however, also scales up the volume of data that needs to be analyzed. Our analysis made use of many powerful software tools to convert this large volume of data into meaningful results. We give a brief description of the key hardware and platform resources our simulations required.

Hardware

Our simulations make use of both local and cloud computing resources. While most development and analysis work is completed locally, cloud resources are critical to simulating a wide variety of alternatives and scenarios in a timely manner.

Local resources

Our local computing resources consisted of two Dell PowerEdge T620 servers, each with dual 6-core/12-thread 3.50GHz Intel Xeon E5-2643 v2 processors, and 64GB of 1866MHz RAM.

Cloud resources

Our cloud computing resources are obtained from the Amazon Elastic Compute Cloud (EC2) service. We primarily run our simulations on their compute optimized C3 instances; these provide virtual machines running on servers equipped with 2.80GHz Intel Xeon E5-2643 v2 Intel Xeon E5-2680 v2 processors. Virtual machines are available with the ability to execute a range of numbers of execution threads, and have access to 3.75GB of RAM for every two execution threads.

Software

Our software resources fell into four broad categories: those required to implement our simulation; those required for detailed impairment modeling; those required to scale our simulation up to the cloud; and those required to analyze and assess the large volume of data produced by our simulations.

Simulation platform

Our simulation software uses the PicoSAT solver (see Biere 2008) when determining feasibility of station repacking; it uses Gurobi to solve optimization problems arising when minimizing impairment during the RZR and DRP processes.

Impairment modeling

Generating the impairment data used in our simulations depended on interference data produced via TVStudy software and on geographic data provided by the U.S. Census Bureau. Processing of the TVStudy software inputs and outputs was handled in Python, with extensive use of the Fiona and Shapely libraries for working with geospatial data, and the Pyproj interface to the PROJ.4 library for coordinate system conversions and projections.

Cloud computing

When running simulations in the cloud, we use the StarCluster project from MIT for creating and launching of computing clusters on Amazon's EC2 service. StarCluster provides a distributed, Linux-based computing environment with management and balancing of computing jobs via the Open Grid Engine batch-queuing system.

Post-processing and analysis

The large volumes of data produced by our cloud simulations requires the use of sophisticated tools to produce meaningful results and derive insights. Initial processing of the data is handled via Stata and the Pandas library for Python. Visualization and analysis is largely performed in Tableau Desktop.

Auction simulator

Simulations are critical to our evaluation of scoring rules and other design choices under a wide variety of conditions. We outline here the key details of our simulator. We focus on details or assumptions that went into our particular simulator, and refer the reader to the Comment PN for the full details of the proposed reverse auction design. Many of the key technical details of our auction simulator were involved in our implementations of RZR and DRP, our handling of impairment minimization, and our feasibility checker, each of which we discuss in detail later.

The main logic of the auction simulator is simple. Initially, either the RZR process or the DRP process is run on all auction participants. This determines the initial set of repacked stations, any stations that exited or were frozen by this process, and the prices paid to stations frozen. Once the RZR or DRP process has completed, the main auction simulation begins with the initial repacking and set of active bidders that result from this process. The auction process itself involves repeated rounds of bidding, where each round involves a decrement of the clock, and any station exits or freezes that result from lowering prices. The process of a single round can be seen in Figure A1. This repeated lowering of the auction clock continues until all stations have either exited or been frozen, at which point the simulation concludes.

In order to focus our results on the aspects of the auction we felt most critical to our analysis, we made several key assumptions. First, all of our simulations focused on UHF stations; the interactions and complexity of bids to move to VHF complicate the auction process without much return. Similarly, we assume straightforward bidding on the part of participants: all prices are evaluated simply by comparison to reservation values, and accepted or rejected accordingly if they are above or below those values. Finally, we note that all feasibility checks required during the course of the auction, RZR simulations, and DRP simulations are performed using our feasibility checker (described in detail later).

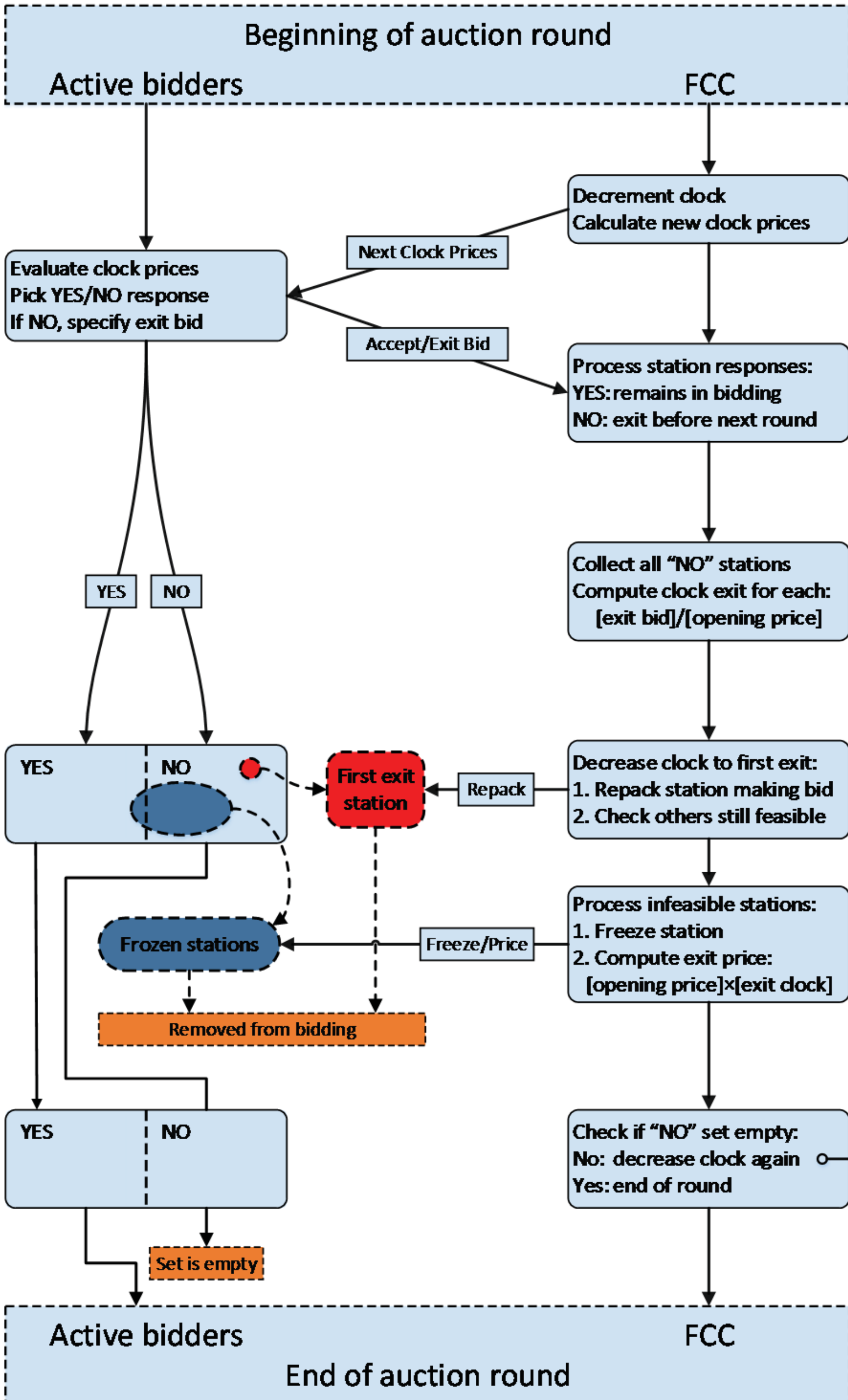


Figure A1: Flow chart of single round of reverse auction simulation

Feasibility checker

Determining the feasibility of repacking stations is a key technical challenge that a successful incentive auction must overcome. This difficulty arises from fundamental computational issues: the repacking problem is readily shown to belong to a class of known hard problems in computer science, which have no known generally efficient solutions. While theoretically intractable, the outlook is fortunately much less bleak in practice. As noted, for example, by Leyton-Brown (2014), the repacking problem lends itself to being expressed as a classical hard problem, namely the Boolean satisfiability problem (SAT). Many heuristics have been developed for the SAT problem that provide efficient solutions in practical applications much of the time. Additionally, solution methods can be tuned to the particular problem instances arising during the incentive auction. For example, both the structure of the interference constraints limiting how stations may be repacked, and the fact that most of the problems we solve are incremental—whether a single station can be added to a set we already know can be repacked—provide natural avenues for improvement.

Improvements such as those mentioned above can boost performance in practice, but they cannot completely avoid the fundamental difficulty of the underlying problem. Despite reducing the complexity of the SAT instances that must be solved, and even completely avoiding the need to solve a SAT problem in some cases, it is inevitable that not all instances will admit efficient solutions. In the context of the auction, choosing to repack a station is a commitment to assign it a channel once the auction completes, and so whenever we cannot efficiently determine the feasibility of a station we must assume the worst and consider it infeasible to repack. As such, the goal becomes to develop a feasibility checker that runs efficiently while solving as many of the problems presented to it as possible.

Just as a successful incentive auction must overcome the challenges of the repacking problem, so must any successful simulation of the auction. In the rest of this section, we outline the technical details of how we handle feasibility checking. Our overall approach combines an established SAT solver with several pre-solving routines designed to minimize the complexity and number of problems the SAT solver must handle. We begin by briefly describing the interpretation of the repacking problem as an instance of the SAT problem, and how we solve these instances, and then go on to describe our pre-solving routines. As our pre-solving routines are based on the interpretation of the interference constraints in the repacking problem as a mathematical graph, we discuss this structure before describing the details of the pre-solving routines themselves.

Satisfiability

We will briefly sketch the interpretation of the repacking problem as an instance of SAT. A more complete description of this interpretation can be found in materials from the FCC LEARN Workshops (see, for example, Leyton-Brown 2014). Instances of the SAT problem ask whether a logical statement can be satisfied. In particular, SAT focuses on statements that can be viewed as asking whether a sequence of claims can all be true at the same time, where each claim is that at least one of a specific set of conditions is true. This naturally captures the repacking problem, since a successful repack must:

1. Choose, for every station, a channel consistent with its domain and the clearing target.
2. Choose, for any pair of interfering channel assignments that could be made for two different stations, at least one of these channels to leave open.

Although we do not provide the details of the reduction, the above does lead to a natural formal expression of checking the feasibility of repacking a set of stations as an instance of SAT.

Our feasibility checker utilizes this reduction to express repacking problems as SAT instances. We then use the freely-available solver PicoSAT to determine the feasibility of each problem encountered. As previously mentioned, it is an impossible goal to efficiently solve every instance of SAT, and one must balance resources expended trying to solve instances against quality of solution. Our chosen solver, PicoSAT, provides various parameters for adjusting how much time is spent attempting to determine feasibility before declaring failure. The parameter we use to cut off execution in our simulations is the propagation limit. Although the details of how this parameter affects solver behavior depends on internal aspects of the PicoSAT solver's implementation that are outside the scope of this paper, we use this parameter as it gives each problem instance the most consistent amount of time. The solver performance is shown in Table A1. In this table, "unknown" instances are those which we did not solve because they reached our cutoff threshold before producing an answer; as previously mentioned, when checking whether we can feasibly add a station to our current repack, we must be conservative and freeze the station whenever we fail to determine feasibility, just as we would if the instance proved to be infeasible. We can see that, on average, both infeasible and feasible solutions are found significantly before the average SAT solver cut-off occurs.

Table A1: Feasibility check performance

	Feasible	Frozen	
		Infeasible	Unknown
Average number of solutions	838,410.77	157.59	336.59
Average solution time (seconds)	0.01	0.75	5.70
Max solution time (seconds)	7.71	13.61	13.34

We experimented with various settings of the propagation limit and found that a limit of 10 million worked well for the purposes of simulation. This conclusion is supported by Figure A2, which shows the cumulative number of feasibility checks by solution type, excluding feasible solutions. In this figure, we can see that the solution rate has largely plateaued by the time our execution cut-off begins to have an effect. This motivates our choice of cut-off, as we must consider marginal benefit from a higher cut-off. In particular, since every simulation involves about 8 hundred thousand feasibility checks, even small increases have a large cost in computation time. This cost must be weighed against the amount of extra information obtained. Figure A2 suggests that the additional information of running the feasibility checker for long periods of time results in little additional information. Especially for our purpose of evaluating alternative scoring rules and other design decisions, setting a propagation limit of 10 million appears appropriate. This results in spending about 6 seconds on unknown instances before reporting failure to solve them.

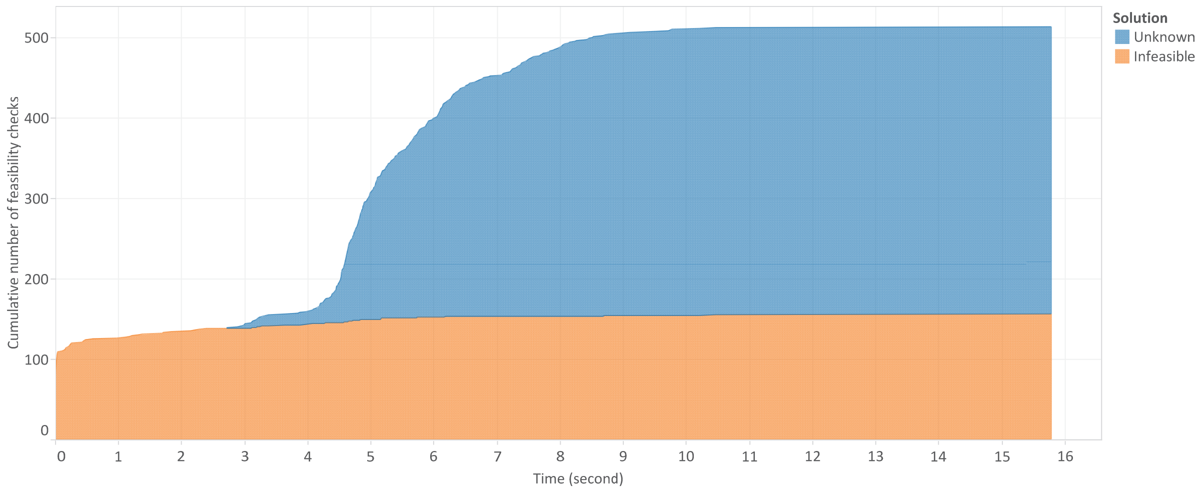


Figure A2: Cumulative number of feasibility checks

Structure of interference constraints

In developing the pre-solving routines in our feasibility checker, we leverage special structure present in the interference constraints between stations. In particular, one can think of interference between two stations as linking them together. Understanding the links of interference between various groups of stations yields critical insights into the structure of the repacking problem. As seen in previous work, this structure can be leveraged to significantly improve feasibility checking routines. Formally, the links that interference creates between stations induces a graph structure on the set of stations, and several key insights into the interactions between stations during the repacking process can be obtained from graph theory and other areas of mathematics.

Locality-based pre-solving

One class of pre-solving routines our feasibility checker implements is based on determining how closely interference links pairs of stations. As suggested by Leyton-Brown (2014), a key observation is that, when packing a set of stations, these interference links identify which pairs of stations directly affect each other when choosing a channel assignment. Furthermore, although stations not directly linked can impact each other’s channel assignments, for this to happen the two stations must be linked by a daisy-chain of intermediate stations, each one directly linked to the last. As noted in previous work, the longer the daisy-chain needed to link two stations by interference, the less likely it is that their channel assignments have any effect on each other. Intuitively, the more links separate two stations, the larger the set of stations that must all be involved if the two are to have an effect on each other. For example, in Figure A3, for station A to affect station E, station C must be involved, whereas for station A to affect station G, at least two additional stations must be involved. When combined with the fact that most of the problems we encounter in the context of the incentive auction ask whether a single station can be added to a repacking, this observation leads to several natural pre-solving approaches that can greatly reduce the size and complexity of the problems that we must submit to the SAT solver.

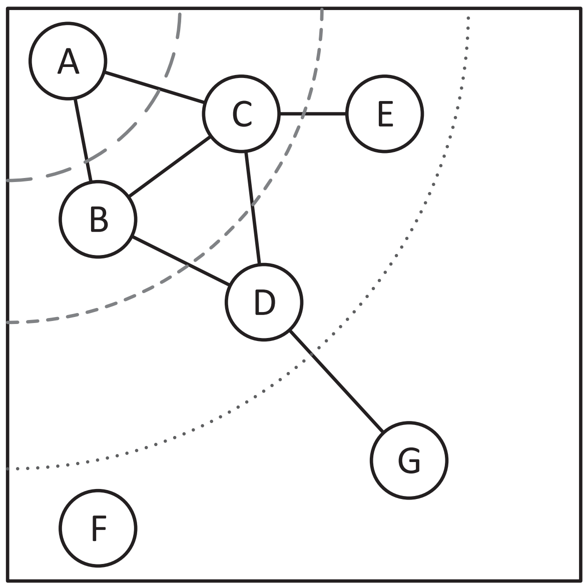


Figure A3: Locality of interference constraints

One immediate improvement that can be made comes from the fact that if two stations are not connected—either directly or by some daisy-chain—through the currently repacked set, the feasibility of repacking one of these stations is completely independent of the other. So if, for example, in Figure A3 stations A, C, and G are currently repacked, we only need to look at stations A and C to evaluate the feasibility of adding station E to this repack set. Thus, when determining whether a station can be repacked, our solver limits the problem to just the stations that are linked by interference to the station in question, either directly or through a daisy-chain. This can lead to significant improvements; for example, the East and West coasts are completely disconnected in terms of interference, and so any repacking of a station on one coast can completely ignore stations on the other.

Another improvement we can make is based on the observation that, the longer the daisy-chain required to link two stations together, the less likely it is that the presence of one will impact the feasibility of repacking the other. Thus, before asking whether it is feasible to pack a station, we ask simpler questions about whether it can be repacked when only considering the stations closest to it. When determining whether a station can be feasibly repacked, we first consider whether we can repack the station itself; the station and those directly linked to it; and the station and those at most two links from it, in isolation from all other stations. For example, in Figure A3, we can see that station A is one link away from each of stations B and C, two links away from each of stations D and E, three links away from station G, and not connected at all to station F. Thus, we would first consider just station A; then stations A, B, and C; then stations A, B, C, D, and E; and finally all stations except station F.

One subtle issue with the above approach is what we can infer about the overall problem from these local problems. In particular, we must be careful about how we formulate these local repacking problems, and about what we infer regarding the overall feasibility of repacking a station from the smaller problems we formulate. The key question is how we handle stations that are outside the local set of our focus.

One approach is to simply ignore them, leaving them out of the repacking problem entirely. Notice, however, that if we determine it is feasible to repack the local set of stations, there is no guarantee that our local channel assignment is compatible with the channel assignment we currently have for other stations. We can, however, be sure that if we cannot repack this small set of stations while ignoring the larger set, there is no hope to find a repacking that works for both the local set and the larger set at the same time. Thus, we can extend an infeasibility result from the local set to the full one, but not a feasibility result.

Another approach is to include the larger set of stations, but keep the problem local by forcing those stations outside our local set to remain on their currently assigned repack channels. This has the opposite problem: any repacking of the local set that we find is compatible with our existing repack by construction. But if we cannot repack the local set there is no guarantee that we could not have done so if we had the flexibility to adjust the channel assignments made for the larger set of stations.

Due to the above considerations, our pre-solving routines try both approaches for each of the local sets we consider. As soon as we see a result that allows us to infer an answer for the full set, we stop our search and report the answer; if none of the local problems yield a general answer—either because they return no answer, or one without implications for the larger set—we then expand our efforts to ask the question for the full set of stations linked by interference to the station we wish to repack.

Clique-based pre-solving

Another class of pre-solving routines that our feasibility checker implements is based on identifying groups of stations that are all strongly linked to each other by interference constraints. Specifically, we are interested in sets of stations where every pair of stations within the set interfere with each other; in the language of graph theory, such a set of stations forms a *clique* with respect to the interference links between stations. Critically, if we have such a set of stations, then each must be assigned a channel different from all other stations in the set, and so we cannot feasibly repack the entire set if there are fewer channels available than there are stations in the set. As Kearns and Dworkin (2014) observed, many infeasible repacking instances can be attributed to the presence of such a set of stations blocking the repacking.

Our pre-solving routines leverage this notion of blocking cliques to identify stations that become infeasible to repack due to such a configuration. In particular, we use the fact that as soon as the number of stations we have repacked in a clique reaches the total number of channels available at our current clearing target, we can immediately declare all remaining stations in this clique as being infeasible to repack. By tracking when such cliques reach their capacity, and identifying the stations that become infeasible when this occurs, we are able to avoid the need to perform any further computation—notably, any need to solve SAT instances—to determine that these stations must be frozen.

One challenge in implementing the pre-solving technique described above is that identifying all of the cliques in a graph is a computationally hard problem. In fact, even the simpler task of determining just the size of the largest clique belongs to the same class of problems as SAT. One approach to surmounting this challenge seen in prior work is to perform random sampling to get a large subset of cliques. We take a different approach and implement the Bron-Kerbosch algorithm for finding all maximal cliques, and in particular the variant proposed by Eppstein et al. (2010). Although in theory the runtime of this algorithm can quite be long, we found that in practice it computes all maximal cliques in the interference graph in a matter of hours. Furthermore, most of this time is spent on identifying small cliques. Since only cliques with more stations than there are channels available under a clearing target are relevant to our pre-solving approach, we prune the search space of the algorithm to remove such small cliques. We found that doing so cut the computation time down to be on the order of minutes.

One subtlety in implementing this pre-solving routine is how the links between stations should be defined, especially since two stations that interfere with each other on one channel may not interfere on a different channel. To avoid the chance of misidentifying stations that are feasible to repack as being blocked by a clique, we use the following criteria: we consider two stations to be linked if, given a particular clearing target, there is no channel available on which both stations can be simultaneously placed. We include both cases caused by co-channel interference, and cases where one (or both) stations are prevented by domain restrictions from being placed on a channel.

While the strength of the above definition guarantees the correctness of all produced results, it brings with it some shortcomings that must be addressed. Most significantly, it ignores the fact that in addition to co-channel interference, it is also possible for stations to experience adjacent-channel interference. Such interference can, in some cases, result in the need to space stations in a clique more widely among available channels. In such cases, we can never simultaneously repack the theoretical maximum number of stations the clique can tolerate. Our pre-solver cannot gain any benefit from cliques where this happens when using the naïve bound on clique capacity; to counteract this, when computing our cliques, we also utilize an optimizer to calculate the maximum number of stations from the clique that can be simultaneously repacked, and use this as the clique's capacity in our pre-solver. Another shortcoming of the approach is that since our definition takes both interference and station domains into account, links between stations are dependent on the clearing target. Thus, we need to compute our lists of maximal cliques once for every possible clearing target.

In implementing this pre-solving routine, we first precompute the full list of cliques induced by interference between stations, for each possible clearing target; for each clique, we use an optimizer to find the maximum number of stations that can be simultaneously repacked from the clique, and save this as the clique's capacity. At the start of a simulation run, the feasibility checker initializes a list of cliques (and their associated capacities) from the appropriate computed list. Over the course of the simulation, the feasibility checker tracks the number of repacked stations in every clique. Each time a station exits and is repacked, every clique it belongs to has its count of repacked stations updated, and for any clique that reaches capacity, any stations belong to the clique that are not already repacked are frozen.

Simulation of initial stages of the auction

The initial stages of the incentive auction play a critical role in determining both how the reverse auction will proceed, and what spectrum will be available for purchase in the forward auction. Here, we discuss two key tasks the initial stages must address: setting a clearing target, and successfully handling situations where there is a lack of competition. We begin with the latter, describing our implementations of the RZR and DRP processes, and conclude with a description of our proposed clearing target selection process.

RZR implementation

Our simulations included a full implementation of our proposed RZR procedure. We refer the reader to Figure A4 for an overview of the RZR procedure itself. Here we describe our implementation of RZR for the purposes of simulation. To put this discussion in context, we briefly outline the steps in the RZR process below.

1. We identify all stations rejecting opening prices, and place them in an initial repack set.
2. We select a clearing target based on this initial repack set.
3. We repeatedly apply the RZR procedure, until either no new stations freeze in round zero, or all stations that do freeze accept the offered RZR price.
4. We determine the final pre-auction state of all participants.

Since our simulations assume straightforward bidding by participants, step 1 is simply a matter of comparing reservation values against opening prices; similarly, step 4 simply requires checking whether each station accepting a RZR prices is still frozen by the final repack, and either making a RZR payment to them or allowing them to continue to the auction proper as appropriate. Step 2 is addressed in a later subsection. Step 3 involves the most complexity, and so we focus on it.

Each round of RZR has three main steps: first, all stations that have rejected either opening prices or RZR prices are repacked, potentially on channels in the 600MHz band as necessary; second, all participating stations that have yet to receive a RZR offer are checked to see if this new repack causes them to freeze in round zero; and finally, all stations newly frozen in round zero are offered their RZR price. The above process continues until we have a round where either no new stations are frozen in round zero, or all stations that are newly frozen accept their offered RZR price.

The main technical challenge in implementing the RZR process comes from finding a repack of the stations which have rejected either opening or RZR prices. Our implementation seeks a repack of the stations that achieves the minimum possible impairment. It does so by formulating an appropriate instance of the optimization program given in Figure A6 and submitting it to the Gurobi optimizer.

Once we have solved our optimization problem and repacked the stations that have rejected offers so far, we proceed to identify any newly frozen stations. We then simulate the process of making each newly frozen station a RZR offer. We handle the process of identifying newly frozen stations via our feasibility checker: we initialize it with the channel assignment found by the Gurobi Optimizer, and then query it to see whether each of the remaining stations can be feasibly added to this set of repacked stations. If not, we must offer the station their RZR price. As our implementation assumes straightforward bidding by participants, evaluating a bidder's response to such an offer is a simple matter of comparing the offered price to the station's reservation value.

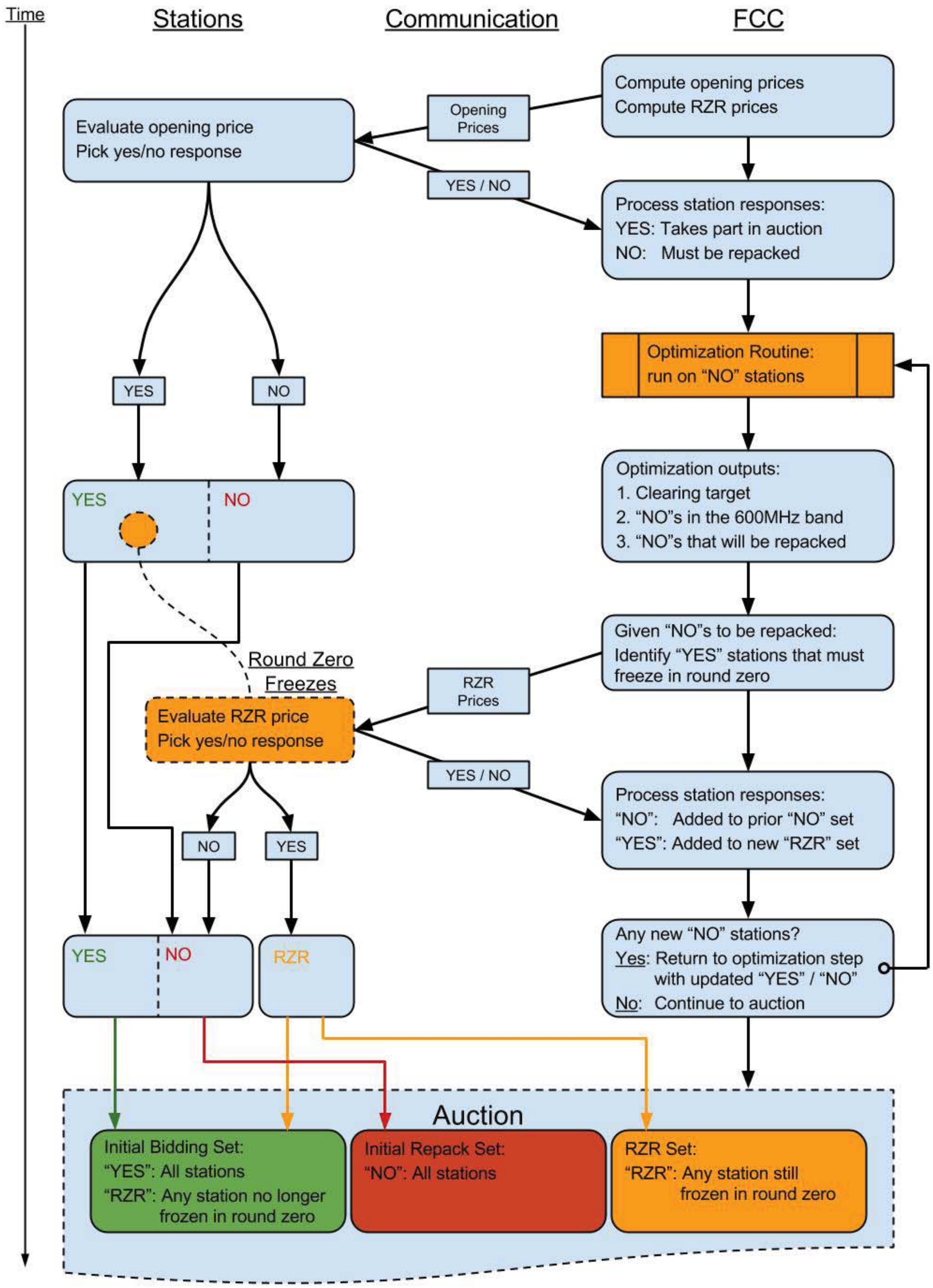


Figure A4: Overview of RZR process

DRP Implementation

Our simulations included an implementation of the DRP procedure proposed in the Comment PN. While we feel our implementation is a natural one, we do wish to highlight that the Comment PN does leave key aspects of the DRP procedure unspecified; this means that there are many valid potential implementations, which may differ substantially from each other. We therefore begin by giving an overview of the DRP process and discussing the key design decisions that must be made to successfully implement DRP. After this, we proceed to describe the design choices we made in our own implementation.

The DRP process splits the auction into two phases: an initial phase in which DRP is said to be "on," and a second stage in which DRP is said to be "off." Once DRP is off, the auction proceeds according to its normal rules. While DRP is on, however, the operation of the auction changes in the following critical way: when a station becomes infeasible to repack, rather than freezing it at its current price, it is instead added to a list of pending freezes and its price is allowed to continue decreasing. The station's price will continue to drop until either it exits, requiring it to be repacked in the 600MHz, or DRP is turns off, causing it to be frozen at its current price at that moment.

The key design decisions in implementing DRP revolve around determining when DRP switches from being on to being off. At a high level, this is decided by comparing the total current and potential impairment—caused by stations that have exited and that are on the list of pending freezes, respectively—to a threshold on the maximum allowable impairment. The key decisions revolve around determining how potential impairment from stations on the list of pending freezes should be calculated, and how the threshold on maximum allowable impairment should be calculated. The Comment PN proposes evaluating potential impairment as the minimum impairment that could result from stations on the pending freeze list exiting, either all stations exiting simultaneously or, alternatively, the single station causing the most potential impairment exiting alone. We focus on the former, as it aligns better with ISIX impairment data.

Calculating the potential impairment from stations on the list of pending freezes is computationally problematic. In particular, minimizing the total impairment caused by repacking a set of a stations is a computationally hard problem, and can require an unpredictable—and possibly quite long—time to solve. This is fundamentally at odds with meeting an active schedule of bidding rounds. On the other hand, while many heuristics can provide approximate solutions to this problem, it is not always clear how close the produced approximations are to the true minimum, or even whether they relate to the true minimum in any sort of consistent fashion. Thus, we are forced by practical considerations to accept unpredictable behavior in either the solution time required or the quality of solutions produced. As a fixed bidding round schedule cannot accommodate unpredictable solution times, it is almost certain we must accept the latter.

While setting a threshold on allowable impairment is in theory more straightforward than computing potential impairment, we note that in fact these issues are inextricably linked. Since it seems certain we must accept an approximate solution to computing potential impairment, the nature of this approximation will determine the meaning of any threshold we might choose. More concretely, say we fix a particular method of computing potential impairment and a particular threshold on allowing impairment. If we decide we want to reduce the typical amount of time that DRP remains in effect, we can either reduce the threshold, or change our calculation method for potential impairment to one that produces, on average, higher numbers. In other words, in practice our bound on allowable impairment is a bound on *approximate* allowable impairment, and the effective bound this translates to on *actual* allowable impairment depends critically on the approximation method used. Thus, we can see that any decisions about impairment thresholds and approximate methods of computing potential impairment cannot be evaluated in isolation.

With the above in mind, we now give an overview of simulating the DRP procedure, and describe the design choices we made in our own simulation. Our implementation of DRP maintains four sets at all times:

- a repack set, containing all station assigned to non-impairing channels;
- an impairing set, containing all stations to be repacked on impairing channels;
- an active set, containing all stations still actively participating in bidding; and
- a pending freeze set, containing all station which remain active but cannot feasibly be added to the repack set.

At all times, we maintain a feasible, non-impairing channel assignment for all stations in the repack set; channel assignments for stations in the impairing set are made independently in each round using a greedy heuristic. The main steps of our simulation are as follows:

1. Initially, identify all non-participating stations, and find a minimum impairment channel assignment of these stations.
2. Add all nonparticipating stations to either the initial impairing set or the initial repack set, based on whether or not the assignment found in 1 placed them on an impairing channel; we add all participating stations to the active set.
3. While DRP remains on, repeatedly find the next station to exit and do the following:
    a. If the station is in the active set, move it to the repack set; find any new stations frozen by this and move them to the pending freeze set.
    b. If the station is in the pending freeze set, move it to the impairing set.
    c. Compute the total existing and potential impairment using a greedy heuristic.
    d. Compare the total impairment calculated in c above and turn DRP off if it exceeds our threshold on allowable impairment.
4. Compute a minimum impairment channel assignment for all stations in the repack and impairing sets
5. Freeze any stations that cannot be feasibly added to the assignment found in 4 at their current prices.

Since we assume straightforward bidding, finding nonparticipating stations and the exit order of participating stations is simply a matter of comparing valuations to prices. All tests for the feasibility of adding a station to the repack set are performed with our feasibility checker. The impairment minimizations in steps 1 and 4 are performed using appropriate instances of the optimization program in Figure A6. Thus, the only technical details left to address are threshold selection and computation of potential impairment.

Our simulations used two different thresholds: a fixed threshold of 20% of the total national weighted population, and a threshold of the initial impairment found in step 1 plus 3% of the total national weighted population.

Our potential impairment calculations used the following greedy optimization heuristic. We begin by fixing the current channel assignments of all stations in the repack set. We then compute the (approximate) potential impairment caused by stations in the impairing and pending freeze sets by repeating the following process. We incrementally construct a channel assignment for the stations in the impairing and pending freeze sets by repeating the following process. For each station in the impairing and pending freeze sets, we consider each channel still available to that station for assignment, and compute the marginal increase in potential impairment adding it to our current assignment would cause. . We then make the channel assignment that, among all of those we considered, caused the minimal increase in potential impairment. While ideally at the end of this process we will have assigned every station in the impairing and pending freeze sets to a channel, this is not guaranteed to be the case: we may make early choices that jointly interfere with all possible channels available to some other station in the impairing or pending frozen sets. In order to achieve as conservative an estimate of the potential impairment as possible, we set the impairment cost of any station left unassigned at the end of this process to be the maximum total impairment it could cause in isolation when assigned to any channel in its domain.

Clearing target optimization

Our proposed procedure for optimizing the clearing target is extremely simple. We identify the largest target that is achieved with minimal impairment in each of the New York and Los Angeles PEAs, and then set the larger of these as our national clearing target. The main detail that must be specified is exactly how we implement finding the maximum possible clearing target in each of these two critical PEAs. The approach we use is based on ISIX impairment data. Note, however, that we manually disallow the clearing targets above 126MHz.

Given a list of nonparticipating stations, we begin by computing the minimum level of impairment we can achieve in each of the New York and Los Angeles PEAs at each potential clearing target. In order to compute the minimum possible impairment in a specific PEA for a given clearing target, we use a modified version of the optimization program in Figure A6. Our modification is quite simple: we assign unit weight to each license for the given PEA, and zero weight to all others. This modified program will tell us the minimum impairment we could ever achieve in the given PEA at a particular clearing target (ignoring the impact on other PEAs). Once we have computed all of the relevant impairment levels, we simply find the largest clearing target at which the impairment level in each of New York and Los Angeles still qualifies as minimal.

The final issue that must be addressed in our proposed clearing target selection process is what level of impairment should qualify as minimal. In early simulations we used a more simplistic model of impairment, and found that a requirement of zero impairment was quite successful; more recent simulations, however, have used the more detailed ISIX impairment data, and we have seen that a strict zero-impairment rule is susceptible to lowering the clearing target significantly due to impairment that does not result in a single non-saleable or category 2 license, and that in fact can be quite close to zero. This indicates it is appropriate to relax this requirement to allow some small amount of impairment; in our most recent simulations, we found that a threshold limiting impairment to be less than the equivalent of 0.3 blocks to give good results (see Figure A5).

Figure A5: Clearing targets selected when requiring at most 0.3 blocks of impairment

Modeling impairment

A key technical challenge in accurately simulating the initial stages of the incentive auction is correctly modeling and calculating impairment arising from the placement of television stations in the 600MHz band. All impairment data used in our simulations was generated by following the ISIX methodology as described in the Comment PN and in the ISIX Second R&O. We focus here on details specific to our implementations, and refer the reader to those documents for a full description of the ISIX methodology.

Generation of ISIX data

The impairment data used in our simulations was generated by replicating the full ISIX methodology. We leave description of this complex process itself to the relevant FCC documents, and focus on implementation details specific to our simulations. All data on television station signal fields, interference, and contours was generated using version 1.3.1 of the TVStudy software; all population and geospatial data came from the US Census Bureau's 2010 TIGER products.

The most difficult impairment case to model is that of wireless base station transmitters causing interference to DTV receivers. This difficulty arises from the fact that the ISIX methodology requires this type of interference be modeled by overlaying the entire country with a grid of hypothetical wireless based stations, placed every 10 km. We note that as neither the Comment PN nor the ISIX R&O specified this grid beyond the spacing, our implementation may use a slightly different grid than the FCC's implementation; however, due to the relatively small granularity of this grid, any differences should have no substantial impact on the resulting data. Finally, we omitted stations from our ISIX computations that our analysis indicated had a freeze probability equal to zero in all clearing targets up to 126MHz. This had no impact on the simulations, as any station which never freezes in any exit order will never need to be placed in the 600MHz band.

The computation required to analyze all of these wireless base stations is substantial, as there are more than 80,000 of them. While the Comment PN references processing interference between television stations and hypothetical base stations by considering all base stations within a 500 km radius of a television station (or a set of nearby television stations), we found that the relatively large number of wireless based stations compared to television stations meant that processing all television stations within a 500km of fixed set of nearby wireless base stations to provide a more manageable division of computation. In particular, we grouped wireless base stations by the state they were located in when running our computations. This difference should have no impact on the final data, however.

Finally, in order to make optimization feasible when minimizing impairment, the ISIX methodology aggregates all data to the county level. The Comment PN suggested a range of possible thresholds for what percentage of the population in a county must suffer impairment before the county as a whole is considered impaired; we chose to be conservative and used the lower end of this range, 10%, when generating ISIX data for our simulations.

Impairment minimization

All impairment minimizations required for our simulations were carried out using full ISIX impairment data generated according to the ISIX R&O. Our optimization program, given in Figure A6, uses the ISIX impairment constraints described in Appendix B, section 5 of the

Comment PN. When running hundreds of simulations, however, this optimization problem becomes a significant computational burden. While optimizing the total weighted impairment is costly, computing the total impairment caused by a particular assignment of stations to channels is quite simple, given ISIX impairment data. Thus, a natural approach is to use a simplified approximation of the constraint data in the optimization itself, and then calculate the true impairment caused by the channel assignment this produces. Our simulations used just such an approach, replacing the county-level impairment constraints in the optimization program with less detailed aggregated impairment constraints for a subset of PEAs. We discuss this modification below.

$$\text{minimize} \sum_{a \in A_\ell} w_\ell p_\ell \text{ subject to:}$$

$$\begin{array}{rccccr}
\sum_{a \in A_\ell} \left(pct^D_{(a,\ell)} y^D_{(a,\ell)} + pct^U_{(a,\ell)} y^U_{(a,\ell)} \right) & \leq & p_\ell & \forall \ell \in L & (1) \\
x_{(s,c)} + x_{(s',c)} & \leq & 1 & \forall \{(s,c),(s',c)\} \in \text{CoPairs} & (2) \\
x_{(s,c)} + x_{(s',c')} & \leq & 1 & \forall \{(s,c),(s',c')\} \in \text{AdjPairs} & (3) \\
\sum_{c \in C_s} x_{(s,c)} & = & 1 & \forall s \in R & (4) \\
x_{(s,c)} & \leq & y^D_{(a,\ell)} & \forall (s,c) \in \text{SC}^D_{(a,\ell)}, a \in A_\ell, \ell \in L & (5) \\
x_{(s,c)} & \leq & y^U_{(a,\ell)} & \forall (s,c) \in \text{SC}^U_{(a,\ell)}, a \in A_\ell, \ell \in L & (6) \\
y^D_{(a,\ell)} & \leq & y^U_{(a,\ell)} & \forall a \in A_\ell, \ell \in L & (7) \\
p_\ell & \leq & 1/2 + N_\ell/2 & \forall \ell \in L & (8) \\
p_\ell & \geq & N_\ell & \forall \ell \in L & (9) \\
x_{s,c} & \in & \{0,1\} & \forall s \in S, c \in C_s & (10) \\
p_\ell & \in & [0,1] & \forall \ell \in L & (11) \\
N_\ell & \in & \{0,1\} & \forall \ell \in L & (12) \\
y^D_{(a,\ell)}, y^U_{(a,\ell)} & \in & \{0,1\} & \forall \ell \in L & (13)
\end{array}$$

Figure A6: Optimization program for impairment minimization using ISIX data

Our modification arose from two considerations: first, that a full optimization introduced an impractical computational burden to the task of simulating the auction in full; and second, that the majority of weighted population is concentrated in a very small number of PEAs. For example, the top 5 PEAs (by population) account for over 40% of the total national weighted population, while the top 35 account for over 75% of the total. Thus, we considered approaches that model the impairment in top PEAs (by population) exactly, while approximating the impairment in PEAs with smaller populations.

Before we discuss our modifications to the optimization program, we briefly review the variables and sets involved in the original optimization program given in Figure A6. Since the constraints feature only minor differences from those given in Append B of the Comment PN, we refer to the reader to that document for a full discussion of the motivation and intuition behind the various constraints, and just review the definitions necessary to the optimization program. The optimization program depends on several sets:

- S is the set of all stations, C is the set of all UHF channels, and L is the set of all wireless licenses available at the current clearing target;
- $R \subseteq S$ is the set of stations that must be repacked;
- $C_s \subseteq C$ is the set of channels available to station $s \in S$ (i.e. its domain);
- $\mathrm{CoPairs}$ and $\mathrm{AdjPairs}$ are the sets of all co-interfering and adjacent-interference pairs of channel assignments, respectively, restricted to the repack set R;
- A_ℓ is the set of counties contained in the PEA covered by license $\ell \in L$; and
- $\mathrm{SC}^D_{(a,\ell)}$ and $\mathrm{SC}^U_{(a,\ell)}$ are the sets of channel assignments $(s, c) \in S \times C$ which cause impairment to license $\ell \in L$ in county $a \in A_\ell$.

Given the above sets, we now define the constants and variables used in the constraints of the optimization program specified in Figure A6:

- w_ℓ equals the total weighted population in the PEA for license ℓ, computed by multiplying the population of the PEA by a PEA-specific index value, both as given in Appendix F of the Comment PN;
- $pct^D_{(a,\ell)}$ and $pct^U_{(a,\ell)}$ are constants representing the portion of the population covered by license ℓ that is contained in county a, evenly divided between uplink and downlink;
- $x_{(s,c)}$ is a binary indicator variable for whether station s has been assigned to channel c;
- $y^D_{(a,\ell)}$ and $y^U_{(a,\ell)}$ are binary indicator variables for whether the downlink or uplink portion, respectively, of license ℓ has been impaired in county a;
- p_ℓ is a continuous variable representing the percentage impairment of license ℓ; and
- N_ℓ is a binary indicator variable for whether license ℓ has been impaired to an extent greater than the 50% threshold and thus become non-saleable.

The above definitions, along with the optimization program in Figure A6, are sufficient to describe our modifications to the optimization program. A deeper understanding of the constraints themselves may prove useful in understanding our modifications, and a detailed discussion of the motivation behind and intuition for them can be found in the Comment PN.

We now describe the modifications to the optimization program in Figure A6 that we used in our simulations. The modifications are based around splitting the set of PEAs into two groups, based on their population rank: given a predetermined cutoff T, we form a detailed group and an aggregated group, containing PEAs with population rank at least T and population rank strictly less than T, respectively. We redefine L as the set of wireless licenses for detailed PEAs, and define $\bar{L}$ as the set of wireless licenses for aggregated PEAs. Constraints on licenses in the detailed set L remain unchanged; those in the aggregated set $\bar{L}$ are modified as described below.

$$
\begin{array}{rcll}
\bar{y}_\ell^D + \bar{y}_\ell^U & \leq & p_\ell & \forall \ell \in \bar{L} \quad (1') \\
pct_{(s,c,\ell)}^D x_{(s,c)} & \leq & \bar{y}_\ell^D & \forall (s,c) \in \mathrm{SC}_\ell^D, \ell \in \bar{L} \quad (5') \\
pct_{(s,c,\ell)}^U x_{(s,c)} & \leq & \bar{y}_\ell^U & \forall (s,c) \in \mathrm{SC}_\ell^U, \ell \in \bar{L} \quad (6') \\
\bar{y}_\ell^D & \leq & \bar{y}_\ell^U & \forall \ell \in \bar{L} \quad (7') \\
\bar{y}_\ell^U, \bar{y}_\ell^D & \in & [0,1] & \forall \ell \in \bar{L} \quad (13')
\end{array}
$$

Figure A7:
Modified constraints for aggregated PEAs in approximate optimization program.

For each aggregated wireless license $\ell \in \bar{L}$, we begin by aggregating relevant sets and variables. We only describe these modifications for the downlink portion of licenses, but the modifications for the uplink portions are symmetric. First, we replace the set of binary county-level impairment variables $y_{(a,\ell)}^D$ with a single continuous PEA-level impairment variable $\bar{y}_\ell^D$. Second, we union the sets of impairing channel assignments for each county in A_ℓ to create the single combined set $\mathrm{SC}_\ell^D = \bigcup_{a \in A_\ell} \mathrm{SC}_{(a,\ell)}^D$ of aggregated PEA-level impairing assignments. Finally, for each of the impairing assignments $(s,c) \in \mathrm{SC}_\ell^D$, we sum all of the individual county-level impaired population percentages associated with this particular assignment to obtain a total impaired population percentage $pct_{(s,c,\ell)}^U = \sum_{a:\,(s,c) \in SC_{(a,\ell)}^U} pct_{(a,\ell)}$. Using the above definitions, we can now define the new constraints for our optimization program. For each of the aggregated licenses $\ell \in \bar{L}$, we relax each of the constraints (1), (5), (6), (7), and (13). We give these relaxations in Figure A7. The intuition for each of these changes is as follows:

- the relaxed constraint $(1')$ enforces that impairment to license ℓ is (at least) the sum of the impairments to its uplink and downlink portions;
- the relaxed constraint $(5')$ and $(6')$ enforce that the downlink and uplink portions of license ℓ, respectively, are impaired to at least the extent that any single channel assignment (s,c) impairs each of them;
- constraint $(7')$ enforces that the uplink portion of license ℓ is impaired to at least the same extent as the downlink portion; and
- constraint $(13')$ relaxes the PEA-level impairment variables to be continuous rather than binary.

This completes our modifications to the optimization program.

Our modified optimization program has the following useful features. First, the aggregated PEA-level constraints are significantly simpler than the detailed county-level constraints, leading to practical runtimes for the purposes of simulation. Second, observe that for any feasible solution to the original program, we can produce a feasible solution to the modified program that retains the same objective value, by appropriately aggregating the values of the $y^D_{(a,\ell)}$ and $y^U_{(a,\ell)}$ variables to produce values for the $\bar{y}^D_\ell$ and $\bar{y}^U_\ell$ variables; this implies that the optimal solution to our modified optimization program is always a lower bound on the true minimum nationwide weighted impairment. Finally, our modification provides a simple, intuitive means of balancing accuracy with performance: by increasing the number of detailed PEAs, one can achieve a better approximation at the cost of increased runtime. We describe the performance of this approach in the next subsection.

Performance

Our goal in using a relaxed version of the ISIX constraints in our optimizations was to achieve results that were as accurate as possible within a span of time that was practical when running hundreds of simulations, each requiring multiple optimizations. A critical question when using an approximation is how to evaluate the quality of produced solutions. In our case, the answer is simple: by design, objective values of our approximate optimization program can never be higher than their counterparts in the exact optimization program; this means any lower bounds found during the optimization process apply to the true optimal solution as well. Furthermore, as previously mentioned there is little computational burden involved in computing the exact impairment caused by any channel assignment. This allows us to compute for any solution to our approximate optimization program the exact impairment it would cause in practice. Thus, if we solve our optimization program, and compare the exact impairment caused by the solution found to the best objective lower bound found, we know this ratio must upper bound the relative error of our solution against the true optimum.

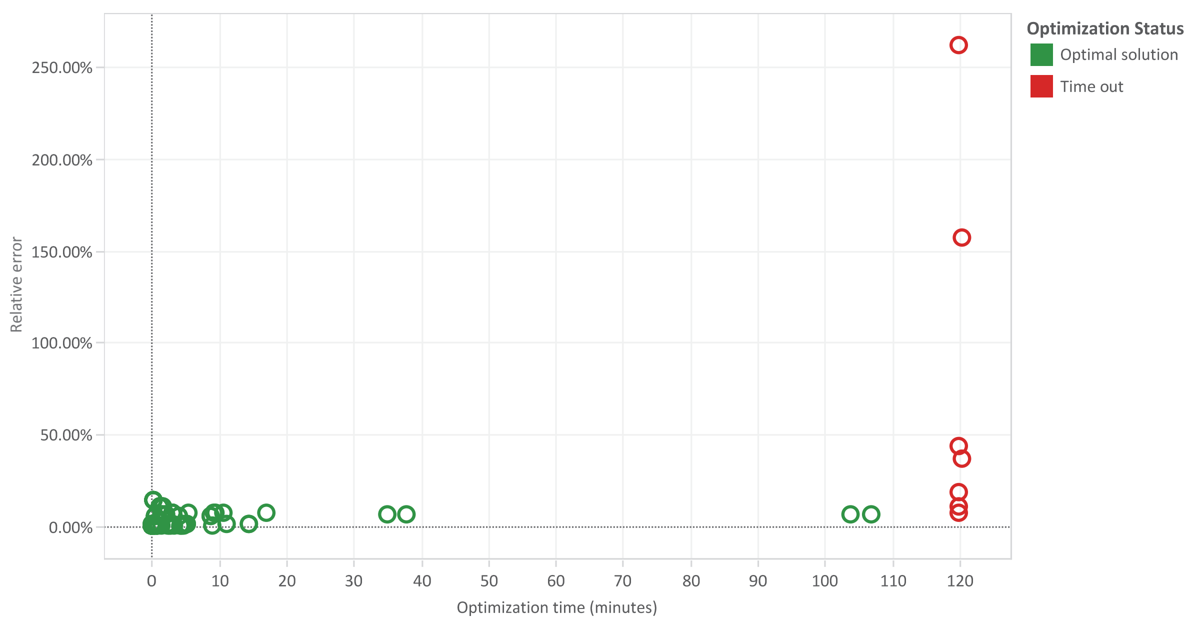


Figure A8: Relative error vs optimization time for initial set of optimizations

Our initial runs of the RZR and DRP processes used an optimization program where the top 35 PEAs by population were detailed, and all others were aggregated (see previous subsection for details). We solved this optimization program with the Gurobi solver on a server with dual 6-core (12-thread) 3.50 GHz Intel Xeon E5-2643 v2 processors, and 64 GB of RAM. Gurobi was configured with a 0.01 optimality threshold and a time limit of 120 minutes, and restricted to use only a single core for each optimization. Figure A8 plots the relative error seen against the optimization time taken. As we can see from the figure, the largest errors arose not as a result of the approximation itself, but as a result of optimizations being halted at suboptimal values due to the time limit.

Based on these results, we re-ran those instances that had one or more optimizations halted at the time limit. We used an even simpler optimization program for these simulations: we restricted the set of detailed PEAs to include only the top 5 by population, using aggregate data for all of the remaining PEAs. We still used an optimality threshold of 0.01 and a time limit of 2 hours for these runs. Figure A9 gives the solution time and relative error rates for the final set of optimizations consisting of these new runs, combined with all of the original runs that did not have any of their optimizations run out of time. Nearly 85% of our optimizations achieved answers with a relative error rate of 1% or less; given that the optimality threshold used in Gurobi was itself 1%, this indicates a high rate of accuracy. Furthermore, over 95% of our optimizations had relative error of at most 10%. This approach would likely achieve even higher accuracy with further tuning, and allowing the Gurobi solver more computational resources for each optimization.

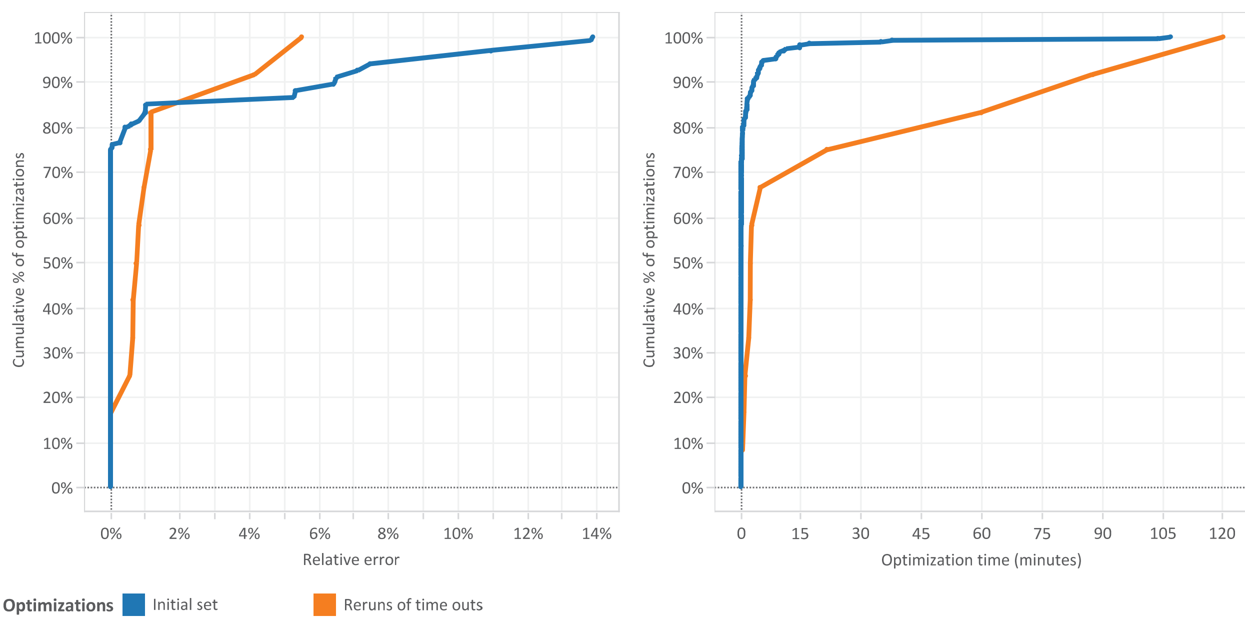


Figure A9: Relative error and runtime rates for final set of optimizations

Scoring rules

The scoring rule determines the opening prices in the reverse auction. These are critical to motivate participation of broadcasters, as the opening price is the maximum price that a station can receive, and a commitment to participate in the auction is a commitment to accept the opening price.

The scoring rule consists of two components, the base clock price and volume, in particular:

$$\text{Score} = (\text{base clock price}) \times (\text{volume})$$

For the base clock price, we considered $1500 in addition to the FCC price of $900. This alternative base clock prices increase the FCC base clock price to encourage participation and thereby make the auction more robust to high broadcaster reservation values.

For volume, we focus on these measures in our analysis:

$$\text{FCC volume} = (\text{Broadcast population})^{1/2} \times (\text{Interference count})^{1/2}$$

$$\text{Reweighted volume} = (\text{Broadcast population})^{1/4} \times (\text{Interference count})^{1/2}$$

where

Broadcast population = a station's interference-free broadcast population (IF). This is the FCC's population measure defined in ¶96 of the Comment PN. We use "broadcast population" rather than "interference-free population" to highlight that this population is referring to broadcast coverage.

Interference count = a count of the station's pairwise interference constraints (IC). This is the FCC's interference measure also defined in ¶96 of the Comment PN.

Following the FCC's approach, we scale all volumes to have a maximum value of one million. This uniform scaling method provides solid ground for comparisons in auction results with alternative volume metrics.

Volume comparisons

In this section we compare the two different volumes under analysis, all scaled to have a maximum value of one million. Figure A10 shows the value of each volume metric. Stations are in a decreasing order based on their FCC volume.

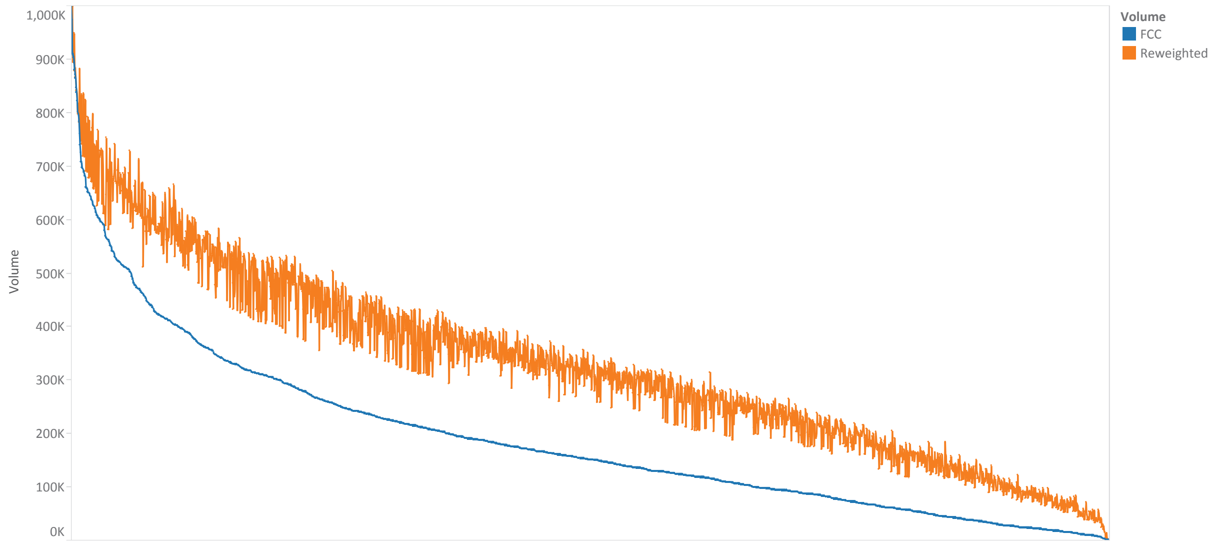
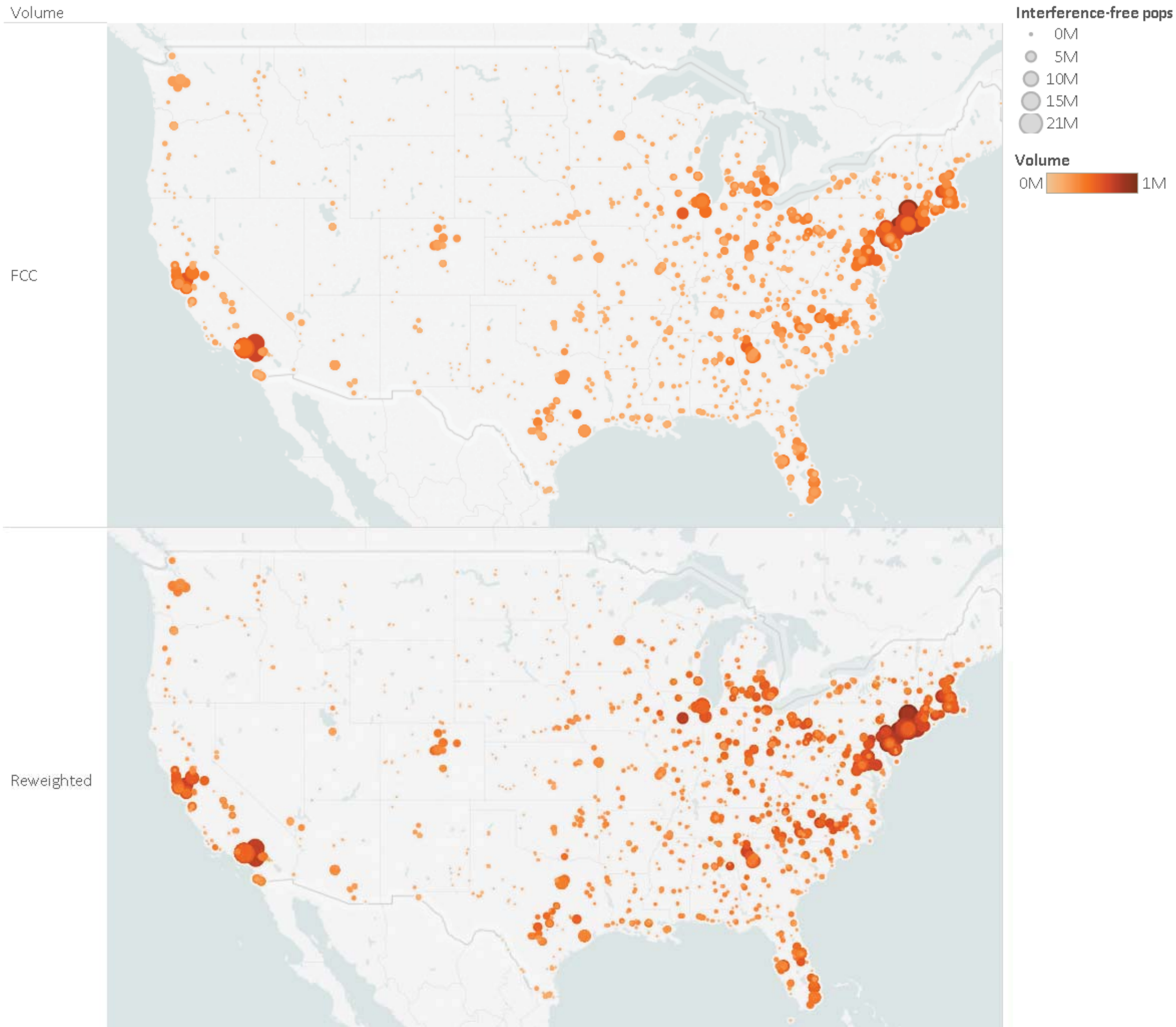


Figure A10: Volume curves

The most important difference among the volumes is the steepness of the volume curve. Reweighted is much flatter than FCC volume. This characteristic cannot be stressed enough. A flatter volume curve means that high-value stations are more apt to be resolved first. Resolution from big stations to small stations tends to promote efficiency, maximizing the value of the repack, as small stations are less apt to get in the way of more valuable larger stations. This intuition is demonstrated in our simulation results. Auctions using Reweighted volume tend to perform better than FCC volume.

Figures A11-12 show each volume metric on a map. Compared to FCC volume, Reweighted volumes have a more gradual transition from high value markets such as New York to mid-value markets in the center of the country.



Figures A11-12: FCC volume and Reweighted volume

Table A2 shows the average opening price for each volume with a $900 base clock price. In top markets the difference between FCC opening prices and Reweighted is relatively small. For example, the differences are 8.13% for New York and 9.86% for Los Angeles. In smaller markets the difference increases, but our simulations show that this does not imply a higher clearing cost since competition drives opening prices down to competitive levels.

Table A2: Average Score for top-20 DMAs

DMA	FCC	Reweighted
New York, NY	711,456	768,812
Los Angeles, CA	564,549	610,417
Chicago, IL	516,713	642,731
Dallas-Ft. Worth, TX	323,069	430,106
Philadelphia, PA	520,556	665,754
Detroit, MI	361,852	516,019
San Francisco-Oakland-San Jose, CA	409,017	550,000
Boston, MA	414,251	572,029
Atlanta, GA	389,383	545,247
Washington, DC	381,212	536,566
Houston, TX	296,842	400,175
Seattle-Tacoma, WA	205,817	318,235
Tampa-St Petersburg-Sarasota, FL	340,409	503,392
Phoenix, AZ	153,237	244,928
Minneapolis - St. Paul, MN	160,000	276,556
Sacramento-Stockton-Modesto, CA	341,597	496,736
Cleveland-Akron, OH	290,864	467,222
Miami - Ft. Lauderdale, FL	266,364	384,091
Orlando-Daytona Beach-Melbourne, FL	298,384	473,081
Denver, CO	190,222	307,911

Precluded population (PC)

The definition of precluded population is the *population that cannot be served by any other station if a certain station is repacked*. It is a quantity that can be derived from the pairwise interference file, together with the associated output from TVStudy. It has many attractive properties, as discussed in the main text. For example,

- For a station that causes no interference, precluded population is its interference-free broadcast population.
- Blocked population is only counted once. Unlike some metrics which grow to large numbers with no intuitive meaning, precluded population produces numbers that still represent real population counts. They are higher than the broadcast population counts because they include blocked populations that are outside a station's service contour or on adjacent channels. So for the KAMU-TV example shown in Table A3, the broadcast population is only 330,386, but the precluded population is 8.5 million. The interpretation of these numbers is simply that if KAMU-TV is assigned to channel 25, it will make it impossible for any other station to provide service on channel 25 to 8.5 million people, including 330,386 inside KAMU's contour, and 8.2 million people outside of KAMU's contour. KAMU should be priced equivalently in the auction to other stations in the same area that block service to 8.5 million people when repacked.

- Simulations show that the sum of the precluded populations of all stations that can be packed onto a single channel across the country averages about 300 million—close to the national population. Intuitively this is right because in a tight repack almost the entire national population should be precluded, otherwise there would be open spaces available for repacking more stations.
- The sum of precluded populations of repacked stations is much less variable in our simulations than the sum of broadcast populations, suggesting it is a better indicator of volume—when optimally packing a trunk with suitcases the sum of the volumes of the packed suitcases is roughly a constant equal to the volume of the trunk.

Precluded population is easily calculated using the following method:

- The FCC paired interference file lists all the stations with which a given station is mutually exclusive ("blocked stations").
- Any point that can receive service from a blocked station, but cannot receive service from any unblocked station is "precluded" from service if the given station is repacked.
- With the detailed cell-level output files from TVStudy, each precluded point can be identified, and the population associated with those points can be added up to determine the total precluded population.
- The same method is repeated for a given channel and each of the adjacent channels.
- We weight adjacent channel preclusion at 50%, because our analysis indicates adjacent channel interference had approximately 1/2 the significance of co-channel interference.

We did this calculation using a proxy channel (25), but this could easily be done on every channel and averaged, or on some other basis to reflect varying preclusion across a channel range.

To get some intuition for the calculation, it is easiest to see an example. The full detail is available in the code and the resulting measure for each station. We focus on KAMU. Table A3 shows the interference free population and the precluded population for KAMU in each PEA using a proxy channel 25. The population measures are then found by summing over all PEAs. KAMU has 330,386 interference free pops and 8.551 million precluded pops.

Table A3: Precluded population by PEA for KAMU-TV (using proxy channel 25)

facid	PEA	IntFreePop	Precluded Pop CO	Precluded Pop Adj+	Precluded Pop Adj-	Total Precluded Pop Count
65301	Houston, TX	12,662	5,821,376	-	-	5,834,038
65301	Austin, TX	11,787	1,085,088	-	-	1,096,875
65301	Waco, TX	-	621,992	-	-	621,992
65301	Nacogdoches, TX	30,794	333,720	44,144	36,914	405,043
65301	Beaumont, TX	-	4,320	-	-	4,320
65301	Victoria, TX	52,077	105,250	189	24	157,434
65301	Eagle Pass, TX	-	225	-	-	225
65301	Bryan, TX	220,054	1,022	-	-	221,076
65301	Brownwood, TX	-	32,309	3,990	1,729	35,169
65301	Corsicana, TX	-	22,209	-	-	22,209
65301	Lockhart, TX	3,012	57,693	-	-	60,705
65301	Jacksonville, TX	-	152	-	-	152
65301	Natchitoches, LA	-	17	-	-	17
65301	Mineral Wells, TX	-	699	-	-	699
65301	Gonzales, TX	-	88	-	-	88
65301	Marble Falls, TX	-	62,051	-	-	62,051
65301	Del Rio, TX	-	14	-	-	14
65301	Lampasas, TX	-	25,078	-	-	25,078
65301	Brady, TX	-	3,897	-	-	3,897
65301	(blank)	-	-	-	-	-
Total		330,386	8,177,200	48,323	38,667	8,551,081

Here are the steps of the calculation. The calculations are done for every 2 km × 2 km cell using detailed coverage data produced by TVStudy. However, the basic logic is easiest to understand graphically.

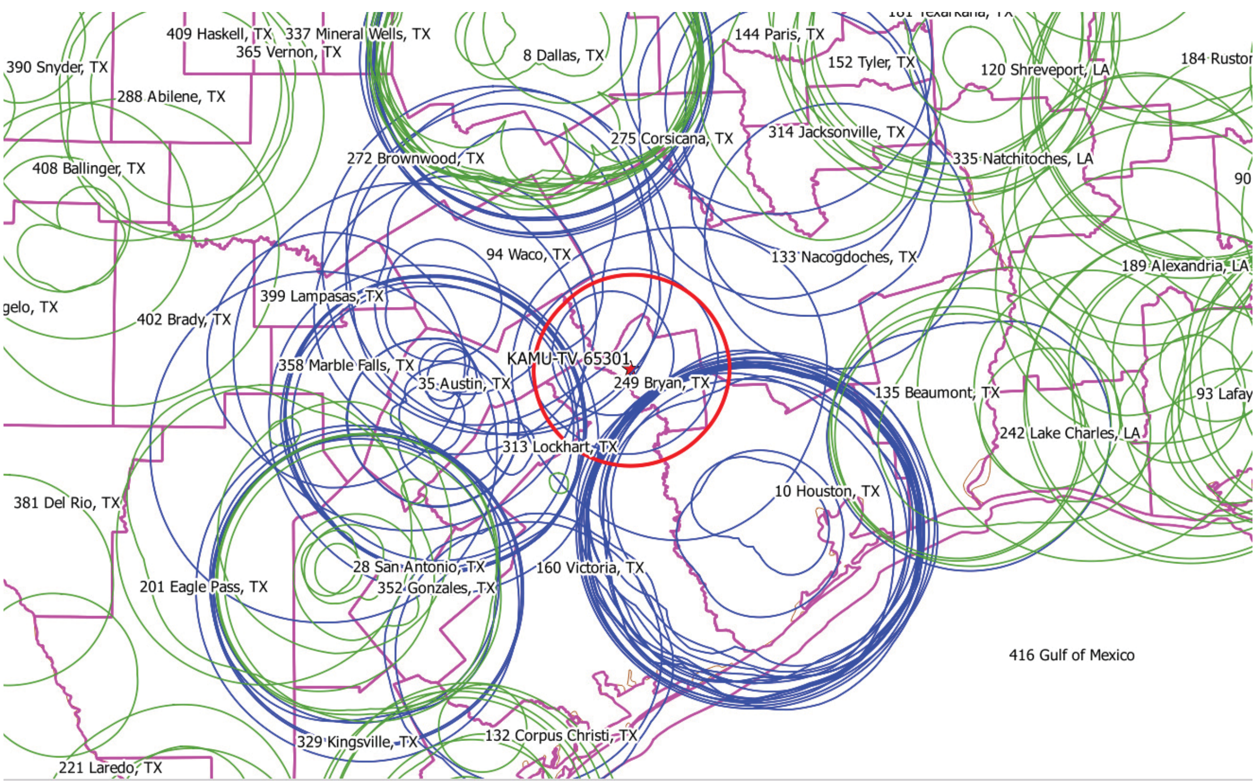
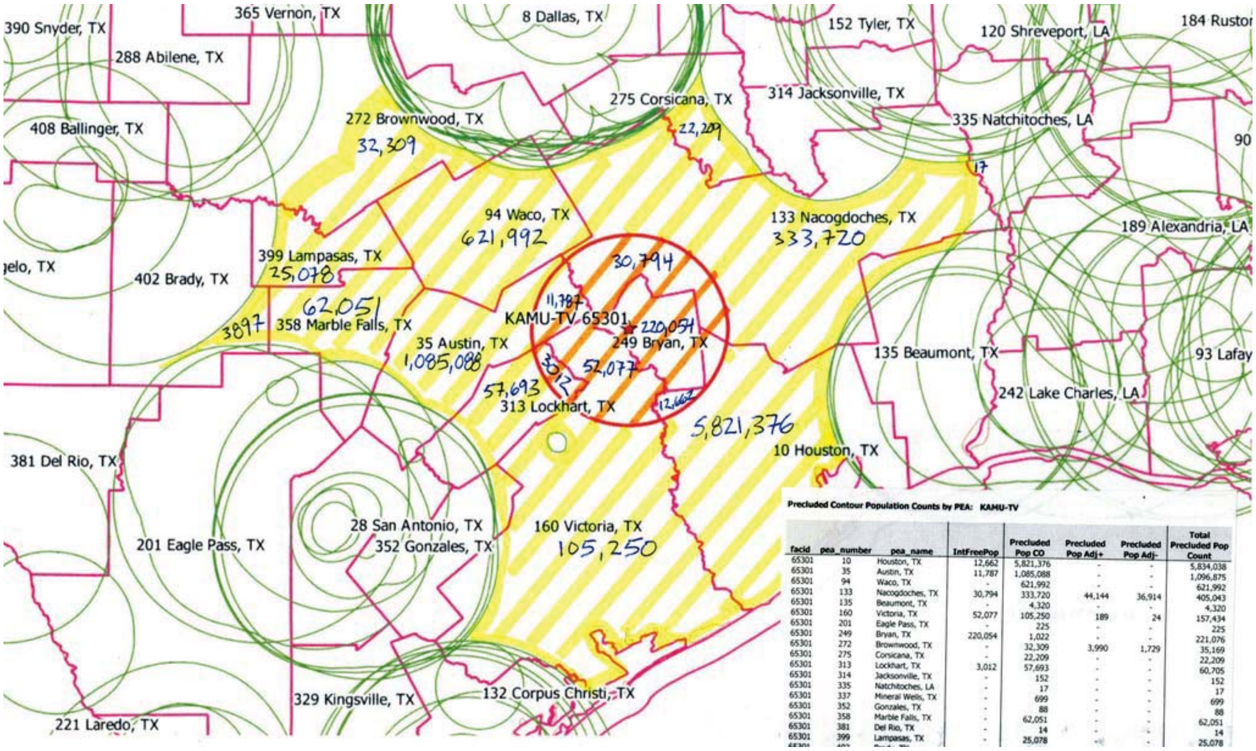


Figure A13: Contour of KAMU (red), contours it interferes with (blue) and does not (green)

Figure A13 shows the first step. For a given station (shown in red), we find all the contours it interferes with (in blue) and all the contours it does not interfere with (in green).



facid	pea_number	pea_name	IntFreePop	Precluded Pop CO	Precluded Pop Adj+	Precluded Pop Adj-	Total Precluded Pop Count
65301	10	Houston, TX	12,662	5,821,376	-	-	5,834,038
65301	35	Austin, TX	11,787	1,085,088	-	-	1,096,875
65301	94	Waco, TX	-	621,992	-	-	621,992
65301	133	Nacogdoches, TX	30,794	333,720	44,144	36,914	405,043
65301	135	Beaumont, TX	-	4,320	-	-	4,320
65301	160	Victoria, TX	52,077	105,250	189	24	157,434
65301	201	Eagle Pass, TX	-	225	-	-	225
65301	249	Bryan, TX	220,054	1,022	-	-	221,076
65301	272	Brownwood, TX	-	32,309	3,990	1,729	35,169
65301	275	Corsicana, TX	-	22,209	-	-	22,209
65301	313	Lockhart, TX	3,012	57,693	-	-	60,705
65301	314	Jacksonville, TX	-	152	-	-	152
65301	335	Natchitoches, LA	-	17	-	-	17
65301	337	Mineral Wells, TX	-	699	-	-	699
65301	352	Gonzales, TX	-	88	-	-	88
65301	358	Marble Falls, TX	-	62,051	-	-	62,051
65301	381	Del Rio, TX	-	14	-	-	14
65301	399	Lampasas, TX	-	25,078	-	-	25,078

Figure A14: Precluded population (co-channel)

Figure A14 shows the second step. We remove all the blocked blue contours, leaving only the green contours that can potentially be repacked co-channel with the subject channel (KAMU). Any population that is inside a blue contour but is not inside a green contour is precluded from service if the station is repacked. The yellow and orange areas are precluded. The orange area is the station's own service area. If the station did not interfere with any other stations, then only the orange area would be precluded. The results can be aggregated in a variety of ways. Below the results are totaled by PEA as in Table A3.

Figure A15: Precluded population (adjacent channel above)

Figure A15 shows the third and final step. The same method is applied to the adjacent channel above and the adjacent channel below (Figure A15 only shows the adjacent channel above calculation). KAMU has little adjacent channel blocking effect. The precluded population calculation includes only ½ of the adjacent channel blocking, because our analysis indicates adjacent channel interference had approximately ½ the significance of co-channel interference.

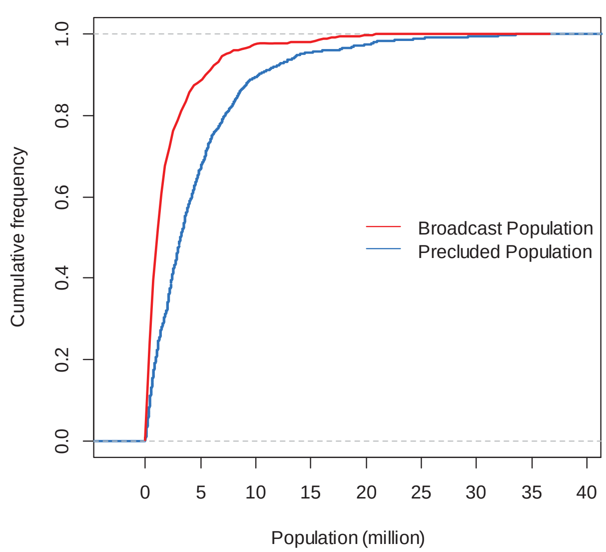


Figure A16: Cumulative distribution of precluded population and broadcast population

Figure A16 shows the cumulative frequency of precluded population and broadcast population. All stations have a higher precluded population than broadcast population, but the difference, as a proportion of the broadcast population, gets smaller as stations with higher populations are considered.

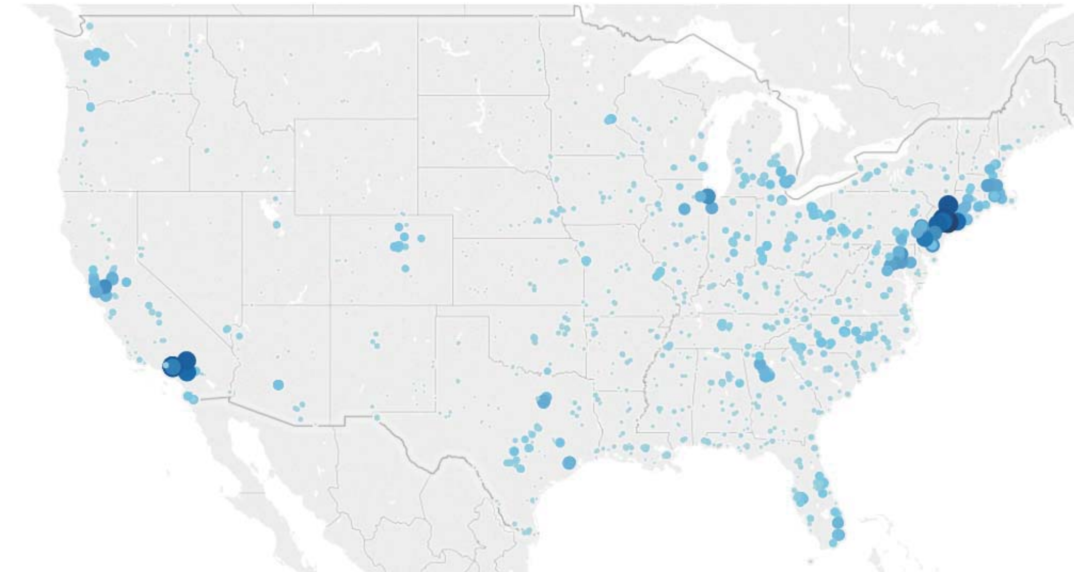


Figure A17: Map of broadcast population

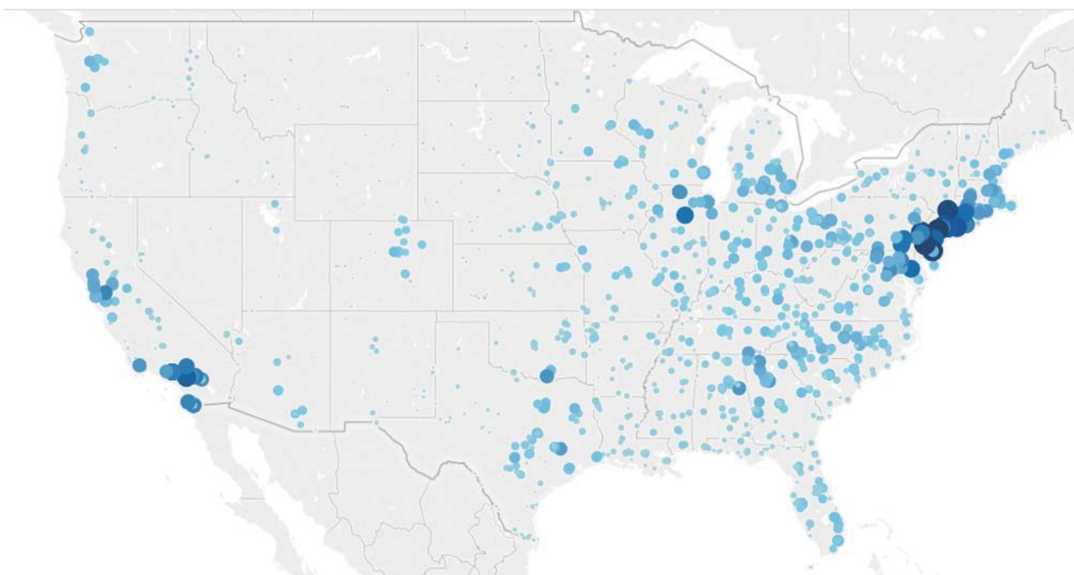


Figure A18: Map of precluded population

Figures A17 and A18 show maps of broadcast population and precluded population. The color and size represent the value for each measure; the larger and darker every mark is, the higher the associated population. Precluded population is more evenly distributed in all markets. This is especially important in New York and Los Angeles since relatively small stations, using the broadcast population metric, are more apt to be repacked, preventing larger stations from serving nearby population.

Freeze probability (FR)

Repacking constraints interact in complex, hard-to-predict ways. Thus, we propose to study the difficulty of repacking stations via an outcome-based, rather than input-based, analysis of the repacking process. We propose the freeze probability (FR), the long-run frequency that a station freezes given a random exit of stations, as a measure of a station's importance in the repacking process. The freeze probability is readily calculated by simulating thousands of auctions with random station exits.

Our calculations (a) assume full participation by all eligible stations; and (b) use completely random exit sequences for the set of stations. This combination of assumptions ensures both that we include the effects of all interference and domain constraints and that our results are free of any bias from a particular valuation model.

We compute the freeze probability as follows. Each simulation follows the same basic steps:

1. Choose a uniformly random order over all UHF stations
2. Begin with an empty repack set
3. For each station in turn check whether it can be feasibly added to the repack set:
    a. If yes, repack the station (add it to the repack set)
    b. If no, freeze the station and leave the repack set unchanged
4. At the end of the above process, every station is either repacked or frozen.

We run the above simulation process for a predetermined number of repetitions. At the end, we compute the fraction of the total number of runs in which the station is frozen; this is the station's freeze probability.

One caveat is that the freeze probability analysis is sensitive to station domains; because of this, the measure varies as a function of the clearing target, and is strongly affected by border constraints. One approach we have tried is considering a domain-free variant of freeze probability, but this faces the challenge that it becomes difficult to capture natural market-level variance in number of available channels—for example, that due to land mobile constraints.

Another caveat is that the freeze probability is more difficult to define over the VHF bands than it is for the UHF bands. Repacking issues can only arise in the VHF band due to relocation of UHF stations, and so it is more difficult to find a simulation-based approach that estimates VHF freeze probabilities while remaining mechanism-free.

We can compute the freeze probabilities of all UHF stations with good accuracy by performing a sufficiently large number of simulations. We bound the maximum error seen across all stations as follows. For each station, freezing is a binary process—it either does or does not happen. Thus, for each station, our set of simulation runs yields a set of identical and independent

Bernoulli trails. To bound the error seen for a given station, we can use classical concentration bounds such as Chernoff. One complication is that we want to bound the *maximum* error across all stations; since whether two stations freeze can be highly correlated, we use a union bound to go from the error of individual stations to this overall maximum error.

We compute freeze probabilities using 11,500 trials. This number produces a maximum error of 2.36% with a confidence of 99%. Figure A19 shows the behavior of the maximum error and its probability for our selected number of trials. The upper bounds shown are all derived using the combination of Chernoff and union bounds discussed above. The (approximate) lower bound is derived by calculating the exact distribution of the maximum error under the assumption that freeze probabilities are independent. While freeze probabilities are not independent, this assumption allows us to get an estimate of where the true maximum error is likely to lie. Figure A20 shows the number of trials required for different error levels at a confidence of 99%.

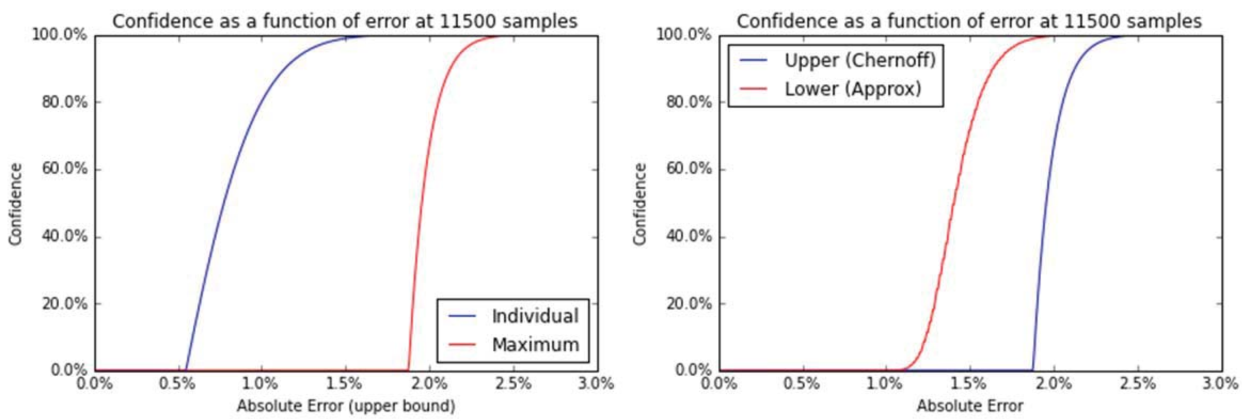


Figure A19: Confidence of FR estimation

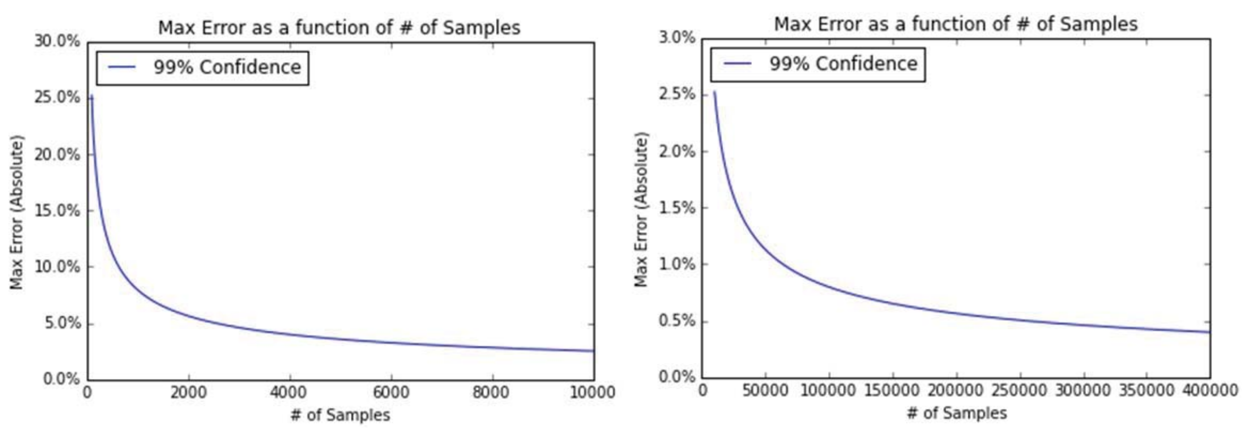


Figure A20: Trials and error of FR estimation

Figures A21 and A22 show maps of freeze probability and interference count. A key difference is that freeze probability correctly identifies the challenges in border markets; whereas, interference count does not. For example, in the Harlingen-Weslaco-Brownsville-McAllen DMA, the interference count is relatively low, despite having a freezing probability of one due to domain constraints.

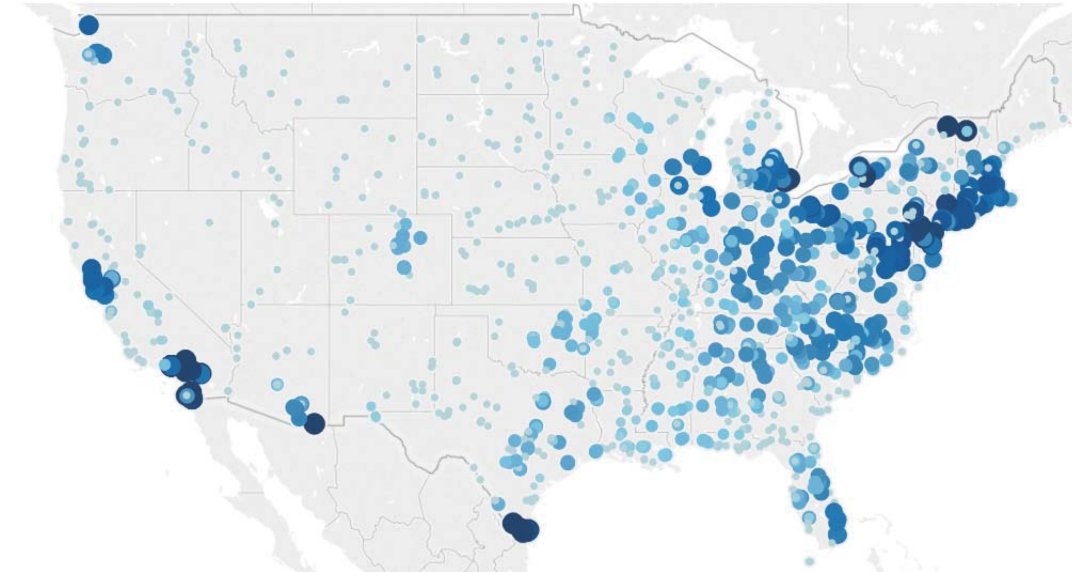


Figure A21: Map of Freeze probability

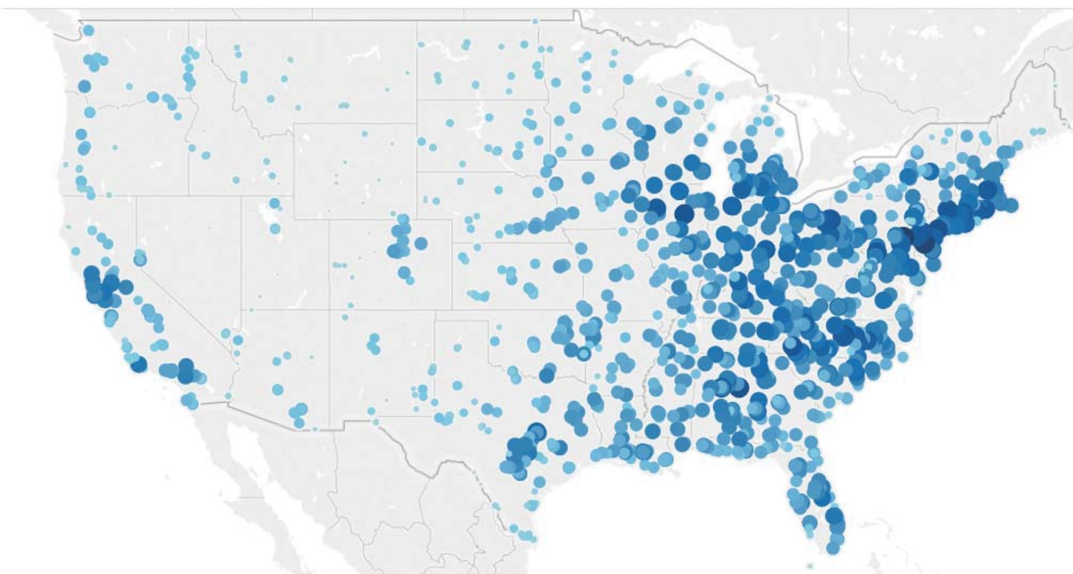


Figure A22: Map of Interference constraint count

When freeze probabilities are used to compute volumes they are bounded below by 0.1 and above by 0.8. The lower bound is introduced to obtain positive volumes for all stations, while the upper bound limits the volumes for stations which freeze in most cases. Figure A23 shows the distribution of freeze probability before and after bounds have been applied. The upper bound applies to a small number of stations.

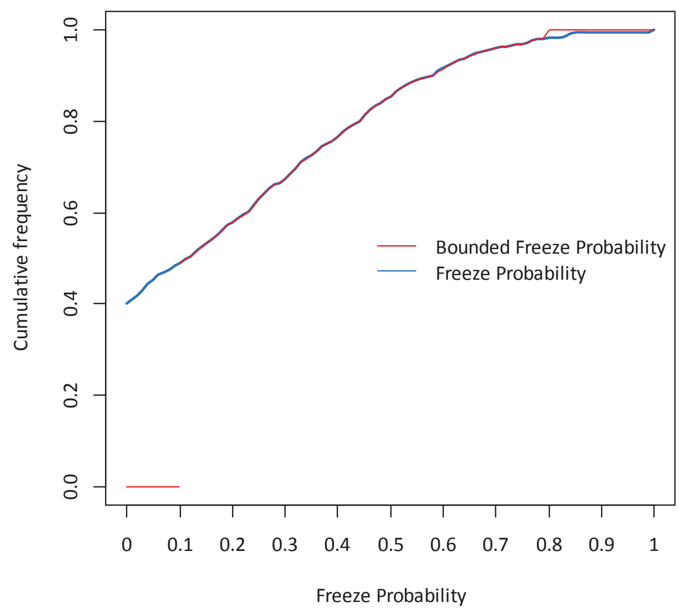


Figure A23: Distribution of Freeze probability with and without bounds

Reweighted volume

The FCC has established in the Public Notice that, to fulfill its mandate, opening prices in the reverse auction will be calculated using a station-specific "volume" factor and an underlying base clock price for a UHF station going off air.

The FCC has proposed to calculate a station's volume using the formula

$$\text{FCC volume} = (\text{Broadcast population})^{1/2} \times (\text{Interference count})^{1/2}$$

Furthermore, the FCC has established that the interference component should measure a station's potential impact on repacking. The FCC proposed the Interference Count (IC) as a measure of interference and Interference-free broadcast population (IF) as a measure of population.

As explained above, the empirical freezing probability (FR) of a station directly measures its impact on repacking. Thus, a volume based on FR would better address the FCC's stated objectives.

Considering that a volume based on FR could be perceived as more complicated than the originally proposed volume, several other alternatives were studied. In particular, we studied variations on the exponents proposed by the FCC.

Improving the FCC volume formula by changing only its exponents have two major advantages:

i) Uses the same inputs (IC and IF), hence minimizing the cost and time of analyzing the change for interested parties, and
ii) Can approximate the effect a station has on repacking.

In order to select which coefficients better reflect the impact each station has on repacking we did a regression analysis on the coefficients of the FCC formula. Specifically we analyzed regressions of the form:

$$\ln(FR_i) = c + a * \ln(IF_i) + b \ln(IC_i) + e_i$$

where FR_i is the empirical freezing probability of station i, IC_i is the Interference count for station i and IF_i is the interference-free broadcast population of station i and e_i is an error. The following table contains basic descriptive measures of each variable. IF and IC measures statistics are reported for the complete set of stations considered by the FCC. FR is reported only for UHF stations.

Table A8: Summary statistics of IF, IC and FR

	Min.	1st Qu.	Median	Mean	3rd Qu	Max
IF	0	379,000	1,039,000	2,088,000	2,428,000	21,190,000
IC	0	41	71	74.49	106	220
FR	0	0.02855	0.24460	0.27180	0.45100	1.0000

The set of station on which the regression is run is of major importance. On the one hand, stations with freezing probabilities close to 0% or close to 100% do not affect the auction outcome. On the other hand, stations with freezing probabilities close to 50% can have significant impacts on auction outcomes.

Specifically, stations with FR=0% can always be repacked, hence the results of the auction do not change if these stations do not participate; if they participate these stations will remain active until they exit the auction. Stations with FR=100% can never be repacked. If one of these stations participates in the auction, it will be frozen for sure and cannot take the place of any station in the repacked set.

Stations with FR away from the extremes of 0 and 1 determine the outcome of the auction as variations in initial score among these stations determine the order of exit and hence change the set of frozen and repacked stations. Thus, there is a trade-off between the quantity and the relevance of the data considered in each regression.

We considered a series of scenarios to determine the appropriate set of coefficients that would achieve a good overall fit. We only consider stations with positive IC and IF in the analysis. Table A9 shows the results of several regressions for different subsets of data. Each scenario only considers stations that satisfy $Min\ FR < FR < Max\ FR$. Regression results are in ascending order by the sum of coefficients, column “a+b”.

Table A9: Selected Regression Scenarios

a	b	a+b	a%	b%	Max FR	Min FR
0.21	0.49	0.70	30%	70%	60%	10%
0.23	0.52	0.75	31%	69%	70%	10%
0.25	0.47	0.71	35%	65%	80%	10%
0.26	0.44	0.71	37%	63%	90%	10%
0.26	0.44	0.71	37%	63%	100%	10%
0.30	0.56	0.85	35%	65%	100%	6%
0.29	0.49	0.77	37%	63%	100%	7%
0.28	0.48	0.76	37%	63%	100%	8%
0.27	0.46	0.73	37%	63%	100%	9%

Based on our findings, we propose to use the following formula for volume:

Reweighted volume = (Broadcast population)$^{1/4}$ × (Interference count)$^{1/2}$

In order to select an appropriate trade-off between the quantity and relevance of stations to be included in the regression analysis we used the following measures for every subset considered:

$Fiterror = \frac{1}{n}\sum(\ln(FR_i) - \ln(PFR_i))^2$, where n is the number of stations, FR_i is the freezing probability of station i and PFR_i is the fitted value.

$Totalerror = \frac{1}{N}\sum(\ln(FR_i) - \ln(PFR_i))^2$, where N is the total number of stations, FR_i is the freezing probability of station i and PFR_i is the fitted value. Stations with $FR = 0$ were assigned a value of $FR = 0.0001$

Figure A24 shows the trade-off between normalized versions of these two measures for all subsets of data considered. Fit error was normalized by dividing Fit error in each scenario by the Maximum Fit error; that is, Fit error of scenario Max FR =100% and Min FR=0%. Total error was normalized by dividing Total error in each scenario by the Maximum of Total error; that is, Total error of scenario Max FR =60% and Min FR=20%.

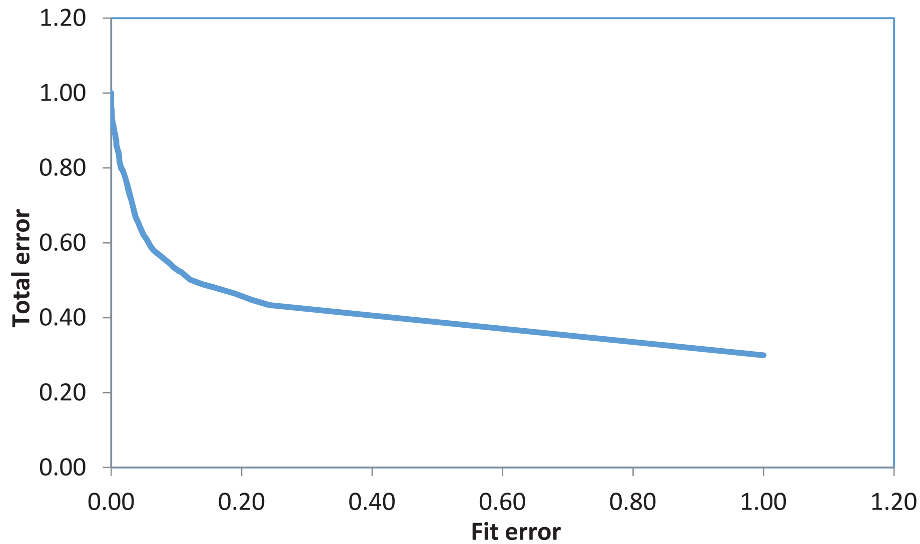


Figure A24: Tradeoff between quantity and relevance

In order to select a best scenario, a point along the curve displayed in Figure A24, we assigned a "price" of one to each normalized error measure; that is, we give them the same weight. We selected scenarios with the lowest "expenditure"—those scenarios that are "tangent" to the line in Figure A24 when equal weight is put on quantity and relevance.

Table A10: Regression Scenarios

a	b	Fit Error	Total Error	nFit Error (1)	nTotal Error (2)	(1)+(2)	Max FR	Min FR
0.295	**0.558**	**0.186**	**9.652**	**0.12**	**0.81**	**0.93**	**100%**	**6%**
0.232	**0.520**	**0.127**	**10.238**	**0.08**	**0.86**	**0.94**	**70%**	**10%**
0.286	**0.487**	**0.165**	**10.039**	**0.11**	**0.84**	**0.95**	**100%**	**7%**
0.282	**0.480**	**0.158**	**10.111**	**0.10**	**0.84**	**0.95**	**100%**	**8%**
0.206	**0.490**	**0.126**	**10.399**	**0.08**	**0.87**	**0.95**	**60%**	**10%**
0.269	**0.461**	**0.146**	**10.307**	**0.10**	**0.86**	**0.96**	**100%**	**9%**
0.247	**0.466**	**0.136**	**10.414**	**0.09**	**0.87**	**0.96**	**80%**	**10%**
0.263	**0.444**	**0.140**	**10.444**	**0.09**	**0.87**	**0.96**	**100%**	**10%**
0.263	**0.444**	**0.140**	**10.444**	**0.09**	**0.87**	**0.96**	**90%**	**10%**
0.256	0.431	0.133	10.571	0.09	0.88	0.97	100%	11%
0.220	0.408	0.088	10.952	0.06	0.91	0.97	70%	15%
0.255	0.410	0.125	10.698	0.08	0.89	0.98	100%	12%
0.252	0.388	0.118	10.835	0.08	0.91	0.98	100%	13%
0.192	0.377	0.085	11.142	0.06	0.93	0.99	60%	15%
0.250	0.368	0.110	10.973	0.07	0.92	0.99	100%	14%
0.235	0.360	0.096	11.119	0.06	0.93	0.99	80%	15%
0.251	0.338	0.100	11.148	0.07	0.93	1.00	100%	15%
0.251	0.338	0.100	11.148	0.07	0.93	1.00	90%	15%
0.242	0.320	0.094	11.330	0.06	0.95	1.01	100%	16%
0.235	0.310	0.091	11.450	0.06	0.96	1.02	100%	17%
0.187	0.331	0.063	11.708	0.04	0.98	1.02	70%	20%
0.228	0.301	0.088	11.571	0.06	0.97	1.02	100%	18%
0.221	0.286	0.083	11.736	0.05	0.98	1.03	100%	19%
0.155	0.293	0.058	11.970	0.04	1.00	1.04	60%	20%
0.204	0.279	0.072	11.897	0.05	0.99	1.04	80%	20%
0.221	0.251	0.076	11.941	0.05	1.00	1.05	100%	20%
0.221	0.251	0.076	11.941	0.05	1.00	1.05	90%	20%

Note: Selected scenarios in bold.

RZR prices

RZR prices are fundamental in determining the quantity and location of impaired licenses to be offered in the forward auction. Stations that are deemed essential to meet a clearing target before the auction begins determine impairment by accepting or rejecting the RZR price offered in the reverse auction. The basic trade-off is that high RZR prices will reduce impairment, but increase the clearing cost somewhat. Clearly RZR prices should depend on carriers' preferences for avoiding impairment.

To balance impairment and clearing costs of round-zero-frozen stations we considered a number of RZR price formulas and settled on one of the simplest approaches. That is, RZR price is a station's opening price multiplied by a multiplier that is less than or equal to 1 and reflects forward auction spectrum value of the particular station. This measure is used to determine the RZR price to be offered to all stations in case they are frozen in round zero.

A station's RZR price is calculated by multiplying a station's opening price by its RZR Multiplier, where

RZR Multiplier = (*Local AWS-3 price*/*Maximum AWS-3 price*)$^{1/2}$

Local AWS-3 price = the weighted average of the prices, in \$/MHzPop, paid in the AWS-3 auction for spectrum in the PEAs that a station's contour touches. The weighting is done on the basis of the interference-free population that the station serves in each PEA, relative to the station's total interference-free population coverage.

Maximum AWS-3 price = the maximum AWS-3 Price in the country, which was \$5.55/MHzPop for Chicago.

Figure A25 shows a map of the proposed RZR price.

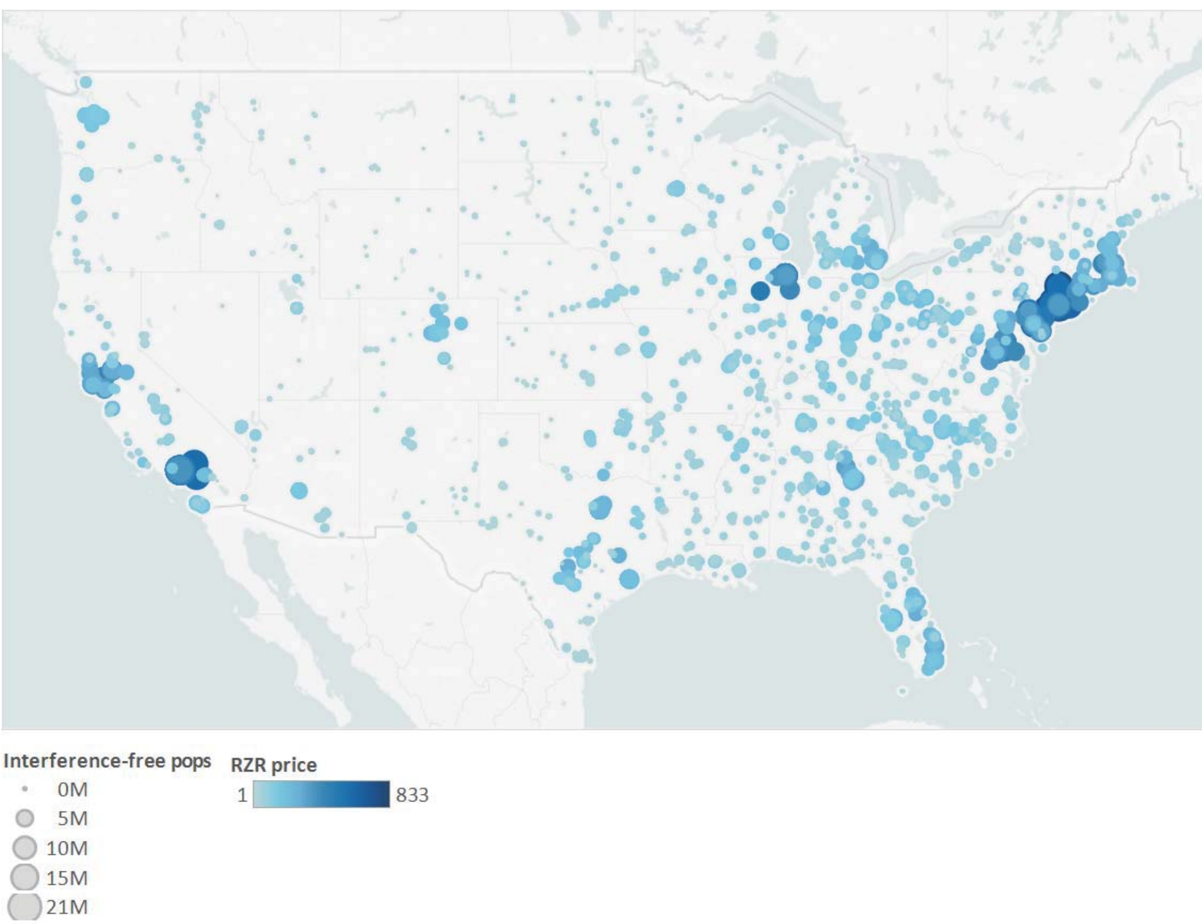


Figure A25: Value-based RZR prices

It is also important to compare RZR prices and opening prices. Figures A26 and A27 do this for the FCC and Reweighted volume, respectively, and a base price clock of $900.

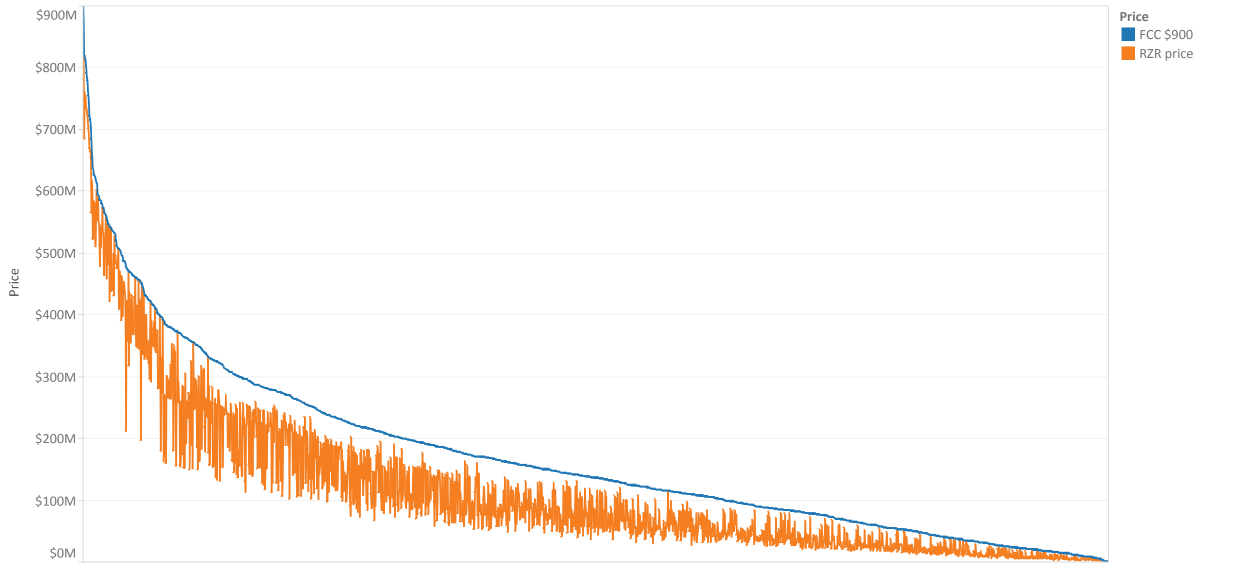


Figure A26: RZR prices for FCC $900

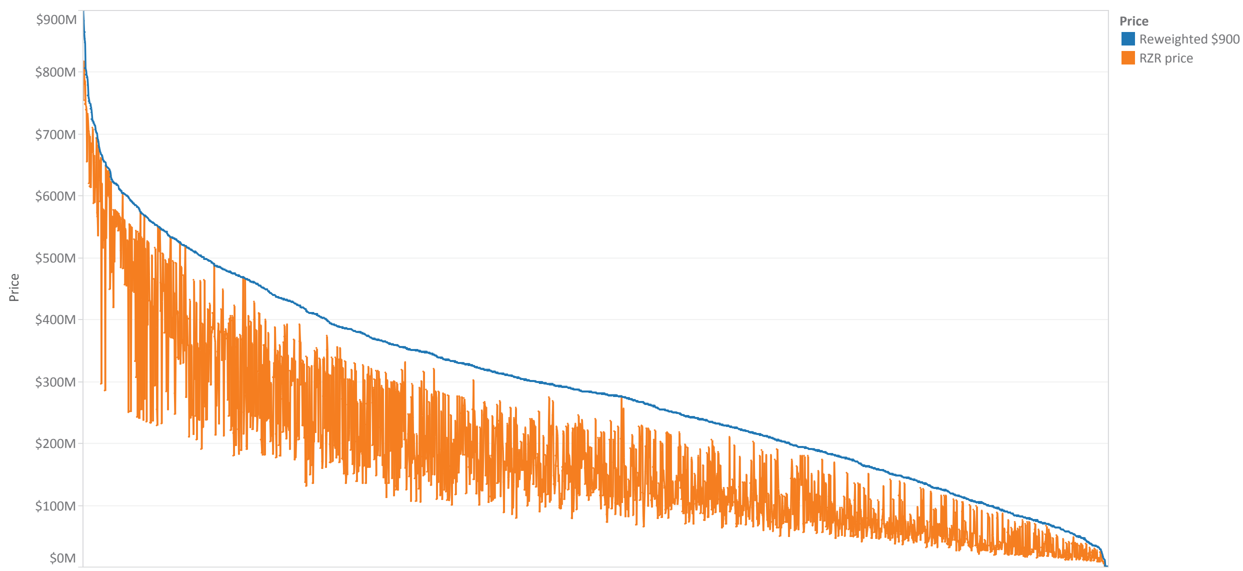


Figure A27: RZR prices for Reweighted $900

Table A11 shows RZR prices for both the FCC volume and Reweighted volume in the DMAs where stations are more likely to be offered a RZR price using a base clock price of $900. However, Value-based RZR prices tend to be higher in most DMAs.

Table A11: RZR prices in DMAs with round zero freezes in million $

		All		RZR Stations	
DMA	RZR frequency	FCC	Reweighted	FCC	Reweighted
San Diego, CA	76.24%	154	255	134	230
Los Angeles, CA	61.73%	525	567	572	612
New York, NY	55.06%	660	713	680	725
Philadelphia, PA	47.82%	422	538	430	551
Detroit, MI	40.57%	257	367	282	393
Laredo, TX	39.81%	22	66	21	63
Wilkes Barre-Scranton, PA	37.44%	131	258	239	423
Palm Springs, CA	36.39%	68	156	99	210
Harlingen-Weslaco-Brownsville-McAllen, TX	34.89%	31	67	31	68
Rochester, NY	25.00%	59	122	62	126
Seattle-Tacoma, WA	20.00%	151	234	108	204
Harrisburg-Lancaster-Lebanon-York, PA	20.00%	248	383	296	458
Burlington, VT-Plattsburgh, NY	19.34%	43	107	29	72
Buffalo, NY	18.52%	56	105	70	127
Cleveland-Akron, OH	13.79%	153	246	178	270
Tucson, AZ	8.72%	55	116	8	30

Note: RZR frequency = likelihood of round zero freeze with value multipliers of 1, 1.5 and 2

Simulation results

We have simulated 180 reverse auction scenarios. These include three variations of the base clock price ($900, $1,250 and $1,500) and two volume metrics (FCC and Reweighted). We also conducted robustness checks, raising reservation values by a multiplicative factor, equal to 1, 1.5, 2, 2.5 and 3, and adding unbiased random error terms to reservation values, using six different random seeds. This section presents detailed simulation results for the base clock prices of $900 and $1,500, and multiplicative factors of 1 and 2.

Figure A28 shows impaired PEA in each scenario. Only PEAs that are impaired in at least one scenario are shown. Of these PEAs, Rochester, Buffalo, Jamestown, NY, Erie, PA and Brownsville, TX are most frequently impaired.

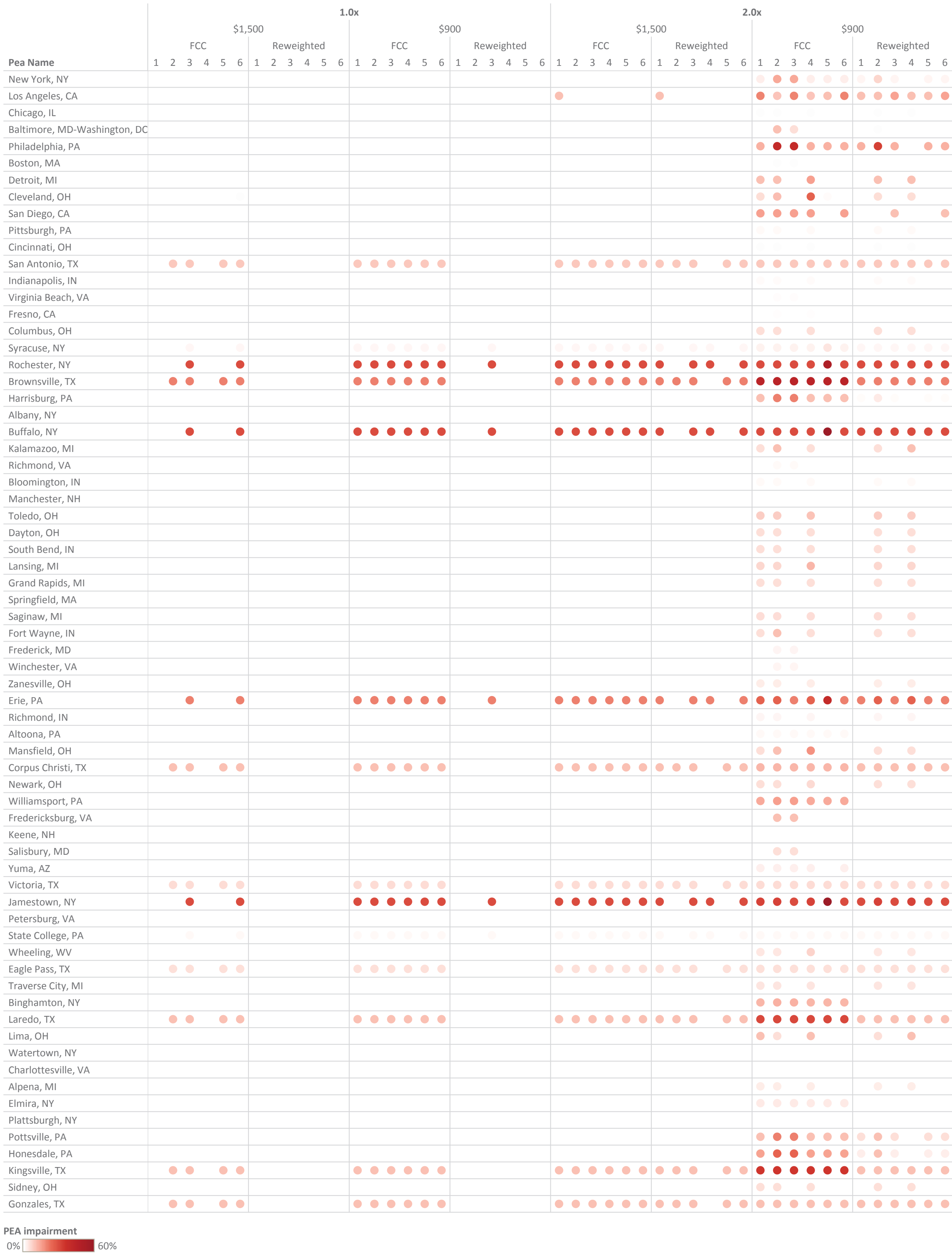


Figure A28: Impaired PEA

In the reverse auction with RZR impairment scheme, each station out of the 1,648 UHF stations in the contiguous United States may be either (1) a non-participant, (2) an impairing station, (3) frozen in round zero, (4) frozen during the auction, and (5) exited during the auction. A non-participant is a station that rejects its opening price or RZR price and it can be repacked. An impairing station is one that rejects its opening price or RZR price but it cannot be repacked. A station is frozen at round zero if it accepts its RZR price. During the auction, a station can either be frozen or exited.

Figure A29 shows the number of stations for each of the five statuses. As the value multiplier rises, the number of non-participants increases and there are more impairing stations or lower optimized clearing target. The higher base clock price can help avoid costly impairments. Even at value multiplier of 2, both FCC and Reweighted volumes with $1,500 base clock price can achieve 126 MHz clearing target with reasonable amount of impairments. Moreover, the Reweighted volume encourages more participations and is thereby less prone to impairments.

Value Multiplier	Base clock price	Volume	Value Seed	Frozen	Exited	Not participate	R0 frozen	Imparing
1.0x	$1,500	FCC	1	493	1,122	19	13	1
			2	506	1,108	20	12	2
			3	503	1,115	16	11	3
			4	502	1,115	17	13	1
			5	491	1,127	16	12	2
			6	498	1,116	20	11	3
		Reweighted	1	508	1,117	9	13	1
			2	498	1,128	8	13	1
			3	503	1,123	8	13	1
			4	501	1,125	8	13	1
			5	505	1,121	8	13	1
			6	505	1,120	9	13	1
	$900	FCC	1	503	1,072	59	10	4
			2	498	1,066	62	18	4
			3	491	1,076	59	18	4
			4	489	1,081	56	18	4
			5	488	1,083	55	18	4
			6	483	1,089	58	14	4
		Reweighted	1	516	1,096	22	13	1
			2	500	1,114	20	13	1
			3	500	1,115	19	12	2
			4	502	1,108	24	13	1
			5	512	1,100	22	13	1
			6	503	1,109	22	13	1
2.0x	$1,500	FCC	1	477	1,058	78	30	5
			2	467	1,052	84	41	4
			3	467	1,058	78	41	4
			4	472	1,049	81	42	4
			5	471	1,050	84	39	4
			6	489	1,058	82	15	4
		Reweighted	1	476	1,103	34	31	4
			2	471	1,100	34	41	2
			3	479	1,104	30	32	3
			4	478	1,101	34	33	2
			5	475	1,100	34	37	2
			6	489	1,111	30	15	3
	$900	FCC	1	440	942	190	64	12
			2	433	947	192	63	13
			3	429	940	199	68	12
			4	429	950	195	62	12
			5	427	948	196	68	9
			6	424	935	206	73	10
		Reweighted	1	449	1,047	81	64	7
			2	441	1,054	79	65	9
			3	446	1,055	72	67	8
			4	464	1,057	80	41	6
			5	449	1,048	76	68	7
			6	451	1,053	75	61	8
				No of stations	No of stations	No of stations	No of stations	No of stations

Figure A29: Station status

Figure A30 shows population coverage loss, clearing cost, and impairment in each scenario. Clearing 126 MHz is achieved in all scenarios. On average, the Reweighted volume reduces viewer loss and creates less impairments than the FCC volume. These significant improvements from Reweighted volume and a higher base clock price increase clearing cost only slightly in percentage terms holding the clearing target fixed.

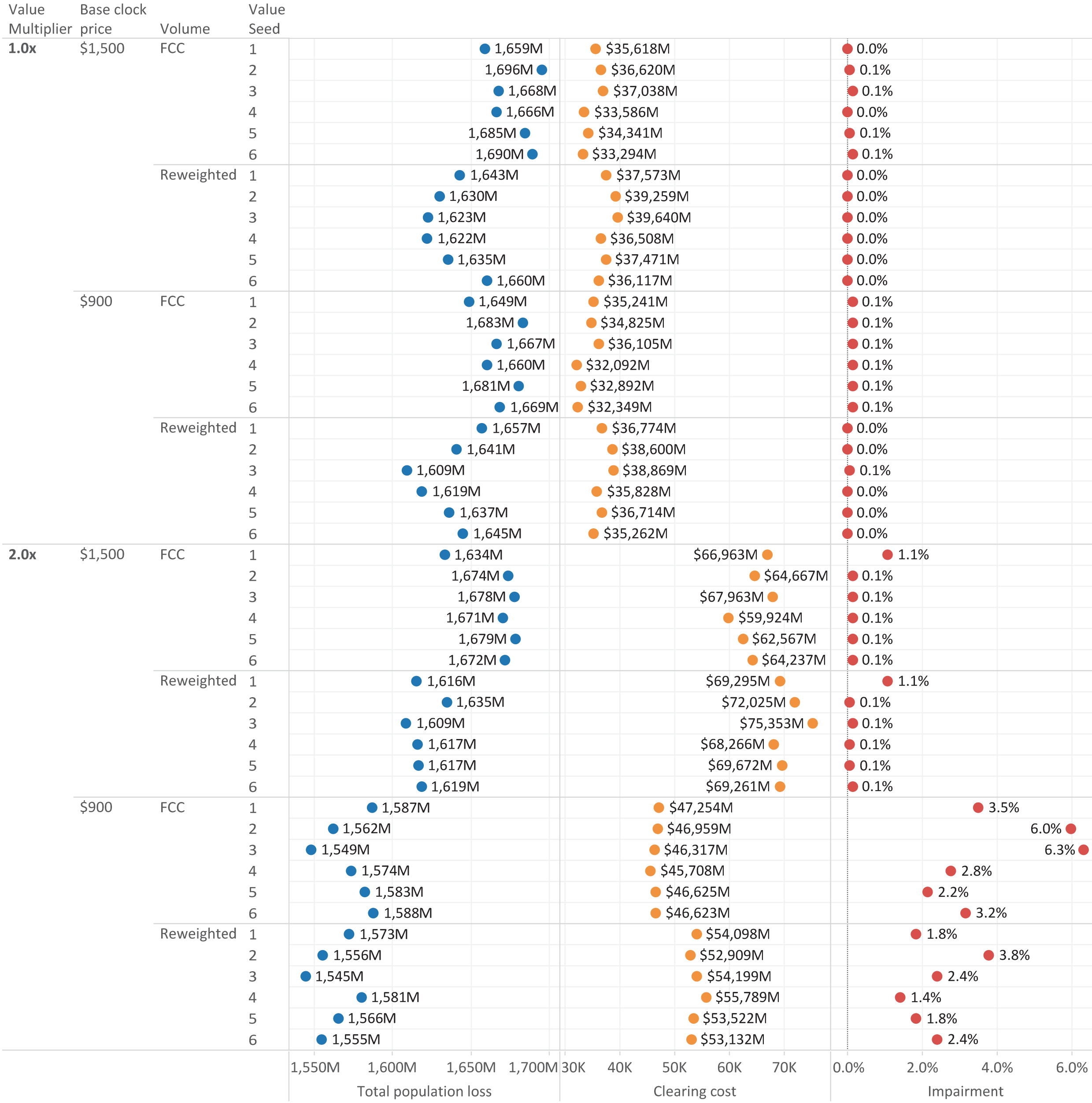


Figure A30: Population loss, clearing cost, blocks cleared, and impairment by scoring rule